Die approbierte Originalversion dieser Diplom-/Masterarbeit ist in der Hauptbibliothek der Technischen Universität Wien aufgestellt und zugänglich.
http://www.ub.tuwien.ac.at

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology.
http://www.ub.tuwien.ac.at/eng

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Multi-Image Morphing: Summarizing Visual Information from Similar Image Regions

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medieninformatik

eingereicht von

Stefan Hödlmoser
Matrikelnummer 0726385

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Martin Kampel, PD Dr.
Mitwirkung: Dipl. Ing. Sebastian Zambanini

Wien, 8.8.2014

(Unterschrift Verfasser)

(Unterschrift Betreuung)

Multi-Image Morphing: Summarizing Visual Information from Similar Image Regions

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Media Informatics

by

Stefan Hödlmoser
Registration Number 0726385

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Martin Kampel, PD Dr.
Assistance: Dipl. Ing. Sebastian Zambanini

Vienna, 8.8.2014

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Stefan Hödlmoser
Christophgasse 1b, 1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasser)

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich während meines Studiums begleitet und unterstützt haben. Ich denke, dass ein Studium nur durch vielfältige, wertvolle Einflüsse von vertrauten Personen gelingen kann:
Vielen Dank an meinen Betreuer Martin Kampel für die fachliche Unterstützung und die Ermöglichung, meine Arbeit am Computer Vision Lab zu verfassen. Ein besonderer Dank gilt auch meinem Zweitbetreuer Sebastian Zambanini für die unermüdliche, stundenlange Unterstützung, wertvollen Diskussionen und die nahezu ständige Verfügbarkeit. Ein herzliches Dankeschön an alle Numismatiker, deren Expertise und bereitgestellten Daten in diese Arbeit einflossen - vor allem unseren Projektpartnern, Klaus Vondrovec und Kathrin Siegl vom Kunsthistorischen Museum Wien.
Weiters möchte ich mich bei meiner gesamten Familie bedanken. Besonders bei meinen Eltern, die mir diese Ausbildung ermöglicht haben. Vielen Dank für die unglaublich hohe Wertschätzung, die ihr mir immer wieder entgegenbringt! Ich möchte mich bei meinem Bruder Michael bedanken für die große Unterstützung während meiner gesamten akademischen Laufbahn und die außerordentliche Verlässlichkeit innerhalb unserer tollen Freundschaft!
Ein besonderer Dank gilt meiner geliebten Freundin Christina für die Bereitschaft, mit mir während meines Studiums durch Dick und Dünn zu gehen und mich in jeder Lebenssituation zu unterstützen und zu begleiten.
Ein großes Dankeschön an all meine Arbeitskollegen der Firma Steinringer Web & IT Solutions - besonders Clemens und Thomas! Mit eurer Hilfe ist es mir gelungen, Beruf und Studium erfolgreich zu vereinen. Zu guter Letzt möchte ich meinen Dank auch an all meine Freunde richten für die humorvollen und ablenkenden Momente in anstrengenden Zeiten.

Danke an euch alle!

Abstract

The process of synthetically producing an image illustrating merged parts of multiple source images is usually known as image morphing. In this Master's thesis a system is developed which morphs more than two source images to one output image. Although the system accepts arbitrary image content, the focus is on using ancient coin images belonging to a common coin type. In ancient times each die was individual in its appearance and had to be renewed manually after several iterations of minting. Nowadays, the coins can be worn or damaged. The goal of the presented morphing framework is the automatic finding and summarization of visual data of common regions by which outliers like wear marks of coins are removed. Since image registration forms the basis of the morphing system, Scale Invariant Feature Transform flow's functionalities are included. The selection of possible region-candidates is decided by exploiting a Markov Random Field in order to find the best combination of visual content. Finally, solving the Poisson equation smooths the morphed image such that any boundaries or disturbing seams become invisible. A twofold evaluation is carried out by firstly applying the system on three different data sets in order to demonstrate visually aesthetic images. Since the aesthetics of an image can be subjective, a second evaluation is done by investigating a classification task of ancient coin images using Scale Invariant Feature Transform flow's energy as similarity metric of visual content. It is shown that substituting a morphed image as training image improves the representation of a coin type compared to a single image. Finally, the thesis is concluded by presenting an analysis about classifying ancient coins with the help of a morphed image.

Kurzfassung

Morphing beschreibt die Verschmelzung von Bildregionen, die von mehreren Eingabebildern stammen können. In dieser Arbeit wurde ein System entwickelt, das die Fähigkeit besitzt, mehr als zwei Eingabebilder zu einem Ausgabebild zu morphen. Obwohl beliebige Bilder akzeptiert werden, liegt der Fokus auf altertümlichen Münzbildern, die alle einen gemeinsamen Typ aufweisen. Zur Zeit der Entstehung dieser Münzen, stellte jeder Münzprägestempel ein Unikat dar und musste nach einer gewissen Anzahl von Prägeiterationen manuell erneuert werden. Heutzutage können altertümliche Münzen abgenützt und beschädigt sein. Das Ziel dieser Arbeit liegt darin, automatisch Münzregionen zu finden, die in allen Eingabebildern vorkommen und diese in einem Ausgabebild zusammenzufassen. Somit werden Ausreißer, wie zum Beispiel Kratzer oder Abnützungen, automatisch aussortiert. Die zugrundeliegende Bildregistrierung erfolgt durch den Scale Invariant Feature Transform Flow Algorithmus. Der Einsatz eines Markov Random Fields trifft eine optimierte Kombination aller möglichen Bildregionen und störende beziehungsweise auffällige Übergänge entfernt die Anwendung eines Poisson-Gleichungs-Algorithmus. Die Evaluierung setzt sich aus zwei Teilgebieten zusammen, wobei im ersten die Flexibilität des Systems demonstriert wird, indem mehrere Datensätze zur Generierung eines gemorphten Bildes verwendet und dessen Ergebnisse präsentiert werden. Im Rahmen des zweiten Teilgebiets wird eine Klassifizierungsmethode vorgestellt, die die Frage beantwortet, wie gut das gemorphte Bild eine Klasse eines Bilddatensatzes repräsentieren kann. Die Energie des Scale Invariant Feature Transform Flow Algorithmus dient dabei als Ähnlichkeitsmaß zweier Bildinhalte. Es wird gezeigt, dass ein gemorphtes Bild dessen Klasse besser repräsentiert als ein einfaches Bild. Den Abschluss dieser Arbeit bildet eine detailierte Analyse der Klassifizierung altertümlicher Münzen mithilfe eines gemorphten Bildes.

Contents

Contents ix

1 Introduction **1**
1.1 Motivation 2
1.2 Contribution 4
1.3 Scope Of Discussion 5
1.4 Results 5
1.5 Structure of the Work 6

2 Related Work **7**
2.1 Summarization of Visual Data 7
2.1.1 Image Morphing 8
Morphing of Two Images 8
Morphing of Multiple Images 12
2.1.2 View Morphing 14
2.1.3 Image Fusion 14
2.1.4 Image Stitching 15
2.1.5 Image Completion 16
2.1.6 Bi-Directional Similarity 16
2.2 Numismatics 17
2.2.1 Making of Ancient Coins 17
2.2.2 Numismatics in Combination with Computer Vision 18
2.3 Summary & Implications for the Proposed Methodology 20

3 Methodology **21**
3.1 Background 21
3.1.1 Image Registration 21
Scale Invariant Feature Transform 22
Scale Invariant Feature Transform Flow 25
PatchMatch and Generalized PatchMatch 26
3.1.2 Optimization 28
Markov Random Field 29
Poisson Equation 30

3.2 Morphing Framework 31
3.2.1 Image Registration 31
3.2.2 Warping 33
3.2.3 Computation and Usage of a Central Image 34
3.2.4 Optimization 36
Find Best Patch Combination Using Markov Random Field 37
Smoothing Disruptive Patch-Boundaries by Solving the Poisson Equation 40
3.2.5 Demonstration on a Synthetic Data Set 41
3.3 Summary 42

4 Experiments **45**
4.1 Data Sets 45
4.2 Qualitative Experiments 49
4.3 Quantitative Experiments 59
4.3.1 Classification Evaluation Schemes 59
4.3.2 Parameter Evaluation by Receiver Operating Characteristic Analysis . . 61
4.3.3 Classification 64
4.4 Discussion 66
4.5 Summary 66

5 Conclusion **67**

Bibliography **71**

Nomenclature **77**

CHAPTER 1

Introduction

Almost parallel to the emergence of this Master's thesis, on an Irish farm a very special animal called *geep* was born. A *geep* is a rare hybrid animal resulted by crossing a goat with a sheep[1]. Very similar to that kind of result in nature, in the field of computer vision the task of 'crossing' differing image contents is called morphing. The morphing process can be expressed in two possible ways.

Firstly, morphing is applied to video sequences where a smooth transformation from one object to another is expressed by multiple generated consecutive video frames as demonstrated in Figure 1.1. With the help of this approach e.g. the Azadi tower of Teheran can be smoothly converted to the notorious Arc de Triomphe of Paris. Some of the first movies which were using this way of morphing as a special effect were *Willow* and *Indiana Jones and the Last Crusade* [62].

Secondly, from several sources a single new image is generated. An example is given in Figure 1.2 where a portrait of the famous scientist Albert Einstein is morphed with a tiger's face. For both ways image morphing comprises some sophisticated approaches where the images' contents are not simply blended but rather their appearance gets aligned to each other by distorting their geometries. As can be seen in Figure 1.2 the eyes of the tiger lie further apart than those of Albert Einstein do. Therefore, a geometric distortion a.k.a. a warp of the image is useful to match the eye's horizontal distance of Einstein and the tiger.

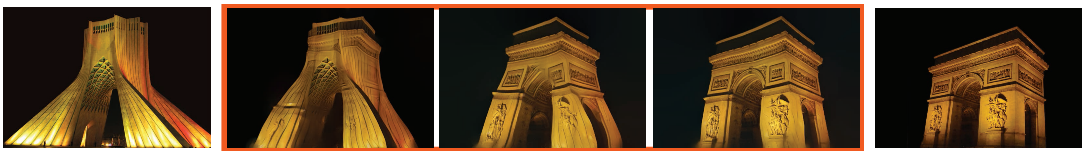
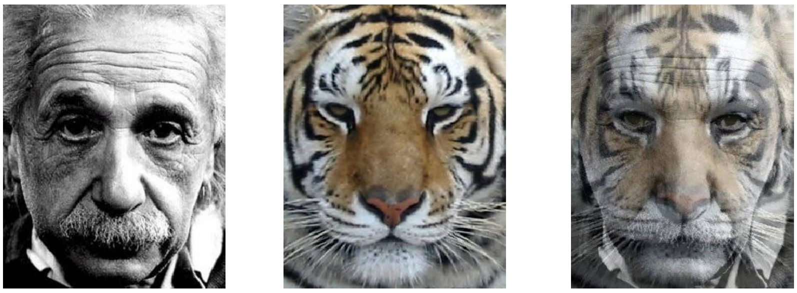


Figure 1.1: A fully morphed transition starting at the source image at the far left to right (figure taken from [15]).

---

[1] `http://www.bbc.com/news/world-europe-26870598` (accessed on 24.04.2014)



(a) (b) (c)

Figure 1.2: A portrait of Albert Einstein (a) and a tiger (b) morphed to a single image (c) (figures available online[2]).

This Master's thesis is about morphing multiple images to one. The majority of the images used show coins originating from the museum of Fine Arts in Vienna. It holds one of the world's five largest numismatic collections where about 700.000 objects are numbered originating from three millenniums. Paper money, medallions, orders and especially coins can be found among these objects[3]. Some of the exhibited coins date from the Roman Republican period, starting around 500 BC, from which overall 550 different coin types arose.

Scientists have made it their responsibility to acquire digital information of the Roman Republican coins by taking photographs of each single coin. Through this work computer aided (and thus automatic) processes like image morphing can be applied to coin images which leads to entirely new insights into the field of ancient coins.

1.1 Motivation

In the years around 500 BC where the Roman Republican period began, the process of minting coins was an individual, manual procedure. Each hand-made coin die was individual in its appearance and had to be renewed due to wear marks after several iterations of minting (about 10.000 iterations) [21]. Dies for the coin's front and back surface (called obverse and reverse) might wear off in varying rates which means that the dies are not only substituted as a pair but rather be replaced individually. Consequently, coins of the same type can have varying combinations of obverse and reverse. Nowadays the coins might be worn or damaged but the degree of preservation can change between individual specimens of a coin type and even locally on a single coin [21]. Figure 1.3 demonstrates how the appearance of the coins differ within the same types. The high variability is demonstrated when considering the writing *VICTRIX* of the illustrated coins. While on the coin at the far left the legend is well preserved, at the far right it is hardly visible. Moreover, the imprint of the second coin (from left) is not exactly centered on the blank leading to incomplete illustrations. Given multiple similar coin images, this work is about automatically finding and summarizing visual data of the coins to a single coin image. Coins are

---

[2] `http://www.morphthing.com/` (accessed on 25.04.2014)

[3] `http://www.khm.at/en/visit/collections/coin-collection/` (accessed on 20.04.2014)

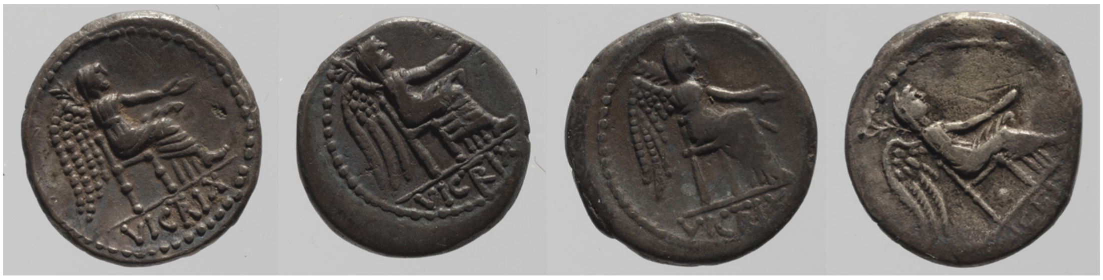


Figure 1.3: Four ancient Roman coins are from the same type. However, the appearance differs from coin to coin.

called similar to each other if they belong to one common coin type. Considering the resulting morphed image the objective is to summarize relevant visual data from each source image. For this thesis visual data of a source image is relevant if it has something in common with other source images. Visual regions which match to several other images are considered as valuable and have to be preserved while non-matchable regions have to be eliminated. With the help of this morphing procedure one single image of a class is obtained which is more representative than any single image of that class.

Morphing has been researched extensively for nearly three decades. Wolberg explains in [63] morphing and its approaches like blending which is the step of interpolating pixel color values. Warping is described as the process of registering the source images to each other in order to geometrically align features of the images. Features are aligned by firstly finding correspondences in image pairs. The correspondences are found either manually by an animator or in an automatic way. Approaches for the manual determination are presented in [5], [35], [34], [36]. For fully-automatic morphing, local image features have to be extracted in order to fulfill a matching of two images. A possible approach is the Generalized PatchMatch algorithm [3], an extension of the initial PatchMatch algorithm [4]. The algorithm is used for determining similar patches in two related image scenes. The most similar patch is called nearest-neighbor and is determined by a dense approximate correspondence search. In [54] and [15] new morphing approaches for two source images are proposed. The morphed image is regenerated from small patches which match patches in the source images. In [15] Darabi et al. show convincing morph results by means of the Generalized PatchMatch algorithm [4]. Another way for finding correspondences is provided by the Scale Invariant Feature Transform (SIFT) flow algorithm, presented in [38]. SIFT flow is based on optical flow methods which are developed to estimate the transitional change of the image's content by comparing intensity values. In other words a dense sampling in time is done which enables tracking of a moving object. SIFT flow uses SIFT features and provides dense sampling in the space of natural images which enables the alignment of images [38]. SIFT flow uses SIFT descriptors [39] to describe extracted features which are compared and rated. Finally, the most similar features form a pixel correspondence.

In this work computed correspondences are exploited for a warping process where the resulting images show one uniform alignment. The objective of outsourcing unitary regions and retaining common regions is fulfilled by introducing a central image whose content can be generated by calculating the mean of multiple source images. Improvements of the resulting visually summarized image are realized by exploiting a Markov Random Field (MRF) which computes the best combination of selected regions. Finally, solving the Poisson equation serves for correcting and smoothing intensity values.

1.2 Contribution

Arranging and acquiring knowledge about ancient coins is a field of research on numismatics. The research area of computer vision aims to interpret and disclose information from image contents. The idea of connecting numismatics with computer vision is realized within the Image-based Classification of Ancient Coins (ILAC) project, supported by Austrian Science Fund (FWF)[4]. Numismatics clearly benefits from computer vision applications [14, 72]. Using the example of the ILAC project, the objective lies in the automatic classification of ancient coins. This can be described as the automatic determination of the coin type seen on the image, based on a number of different types in a database. In total, the contributions of the thesis can be summarized as follows:

- Firstly, the main contribution of this work is to generate a morphed image from more than two images with the help of state-of-the-art methods and applications. Considering coin images, the resulting image acts as a denominator: while wear marks or untypical appearances of coins are neglected, typical visual regions representing the type of the coin are added. In other words the finally morphed image is a representation of a coin type. This thesis is the first work in which several ancient coin images are morphed to a single one.

- Secondly, in order to measure how well the morphed image represents a coin class, a comparison of a traditional classification task and an improved classification of ancient coin images with the inclusion of a morphed image is given. A traditional classification task consists of automatically determining the class of 200 query images. By using the SIFT flow energy it is possible to measure the similarity of two images [72]. All images are classified by determining the similarity for the query image and one training image of each class. The pair showing the lowest energy determines the assignment to the class. In the improved classification process instead of using one single coin image as training image a morphed image of each class is compared to the query image. As a contribution to the ILAC project the automatic classification of the provided ancient coin images is improved by using the morphed image as training image.

- The third contribution is seen from a more general point of view: the morphing framework can not only be applied to ancient coin images but also to arbitrary images. This means, that the proposed method creates a single visual class representation from an arbitrary set of source images.

---

[4] ILAC (FWF:TRP 140-N23), available online at `http://www.caa.tuwien.ac.at/cvl/research/ilac/` (accessed on 03.03.2014)

1.3 Scope Of Discussion

This thesis focuses on the generation of a morphed image from Roman Republican coin image data with more than two source images per class. Thus, the scientific question is: *With the help of computer vision state-of-the-art methods, is it possible to generate a morphed image which represents one class of coins?* This question is of interest for improving an existing exemplar-based classification method using a single image as class reference. Therefore, the morphed image serves as reference image in the classification process.

Additionally, as a part of this work it is shown that the morphing process can be generalized to different image sets. Examples of morphed images generated from different image sets are shown in Figure 1.4.

1.4 Results

Resulting images produced by the proposed system are evaluated in two steps. Firstly, in order to present the system's flexibility according to the image content qualitative experiments are performed on three different data sets: a data set showing airplanes photographed sideways, a human face data set where each face is portrayed with different appearances and a coin data set comprising images of the Roman Republican coinage. For this experiments the output is generated by empirically evaluating the algorithm's parameters in order to obtain a visually coherent and complete image content. In other words it is tried that as much as possible common regions of all source images are summarized and at the same time the morphed image should appear aesthetically without exhibiting any disturbing artifacts. Resulting images of the three data sets are illustrated in Figure 1.4. Image regions which are considered as outliers are outsourced. This behavior is clarified by considering the coin example in Figure 1.4: the first input coin image shows residuals of an old coin design which are consequently removed by the algorithm.

Since the aesthetics of an image can be subjective a second, quantitative evaluation is done. All input images of the morphing framework originate from one common class. This evaluation is based on the principle: *„The better the morphed image, the better is the representation of a common class"*. Therefore, a classification task of ancient coin images using SIFT flow's energy as similarity metric of visual content is combined with the proposed morphing method. In order to classify a query image, for each class a training image is selected. Subsequently, the energies between the query image and the set of training images are determined in order to assign the query image to the class where the energy is at a minimum. With this procedure the suitability of the morphed image for representing the underlying class can be assessed. Using the morphed image as training image raises the question of how to choose the parameter configuration in order to produce an image which performs best for the classification task. For this purpose, parameters are selected according to best performing Receiver Operating Characteristic (ROC) curves.

As a result, the parameter configuration of highly aesthetically morphed images is different to those which perform best in the classification task. Due to these quantitative experiments it can be stated that *the representation of a coin class is better by using a morphed image instead of a single image*. Using a morphed image, a classification rate of 92.5 % can be reached which

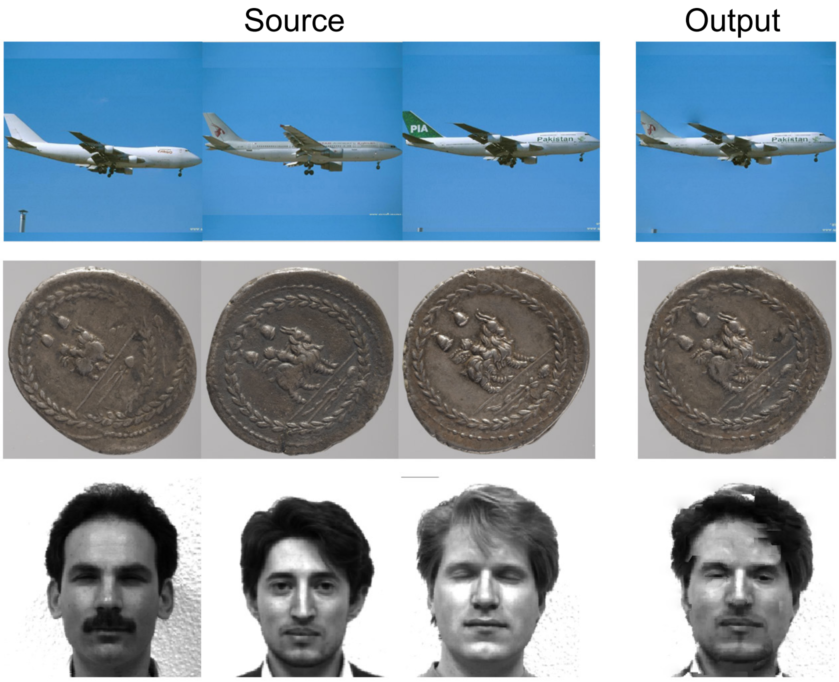


Figure 1.4: Morphed images computed from different data sets.

is an absolute increase of 3 % compared to the usage of a single training image. Moreover, not only the classification rate is increased but also the runtime is reduced by half compared to classification tasks where two training images are used as proposed in [72].

1.5 Structure of the Work

Summarizing visual data from multiple input images requires to examine the relationship of the image's content. Computer vision approaches such as image morphing, view morphing, image completion, image fusion or image stitching share the requirement of registering images to each other. Thus, in Chapter 2 these related approaches of image morphing are investigated. The chapter continues by giving an overview of numismatics, the study of monetary and its underlying medals, coins and related objects.

In order to give a detailed explanation of the implemented morphing framework Chapter 3 introduces the theoretical background, ideas and methodologies on which the morphing framework is based. Thus, the functionalities of optical flow, the SIFT algorithm and the Generalized PatchMatch algorithm are described. Furthermore, this chapter presents the core element of this work - the morphing framework.

All experimental results of this thesis are presented in Chapter 4 which is divided into qualitative experiments and quantitative experiments. Finally, a conclusion comprising a discussion, findings and generalizations of this topic is given in Chapter 5.

CHAPTER 2

Related Work

The topic of this thesis is based on summarization methods of visual information and numismatics in combination with computer vision. This chapter is therefore split into summarization methods of visual data in Section 2.1 and numismatics in Section 2.2. Both of them present related ideas to image morphing, state-of-the-art methods and their applications.

2.1 Summarization of Visual Data

One of the first attempts of image morphing were done by Burt and Adelson in 1983 [11]. They developed an algorithm which summarizes visual data from two or more source images. The algorithm dealt with copy/paste applications which were able to merge different parts of different source images into one plausible and naturally looking target image. Other examples are Szeliski and Shum in [57] who have succeeded in stitching together single images to one panoramic view and Masnou and Morel who introduced in [43] a new algorithm for filling occluded areas in an image with pixels from their neighborhood. The beginnings of these computer vision applications have been further developed. The following research areas deal with approaches for merging different image regions as smooth as possible, making visible and nasty borders between the regions disappear: image morphing blends multiple images into a single one, view morphing produces a synthetic image viewpoint from different views of a scene, image completion fills unknown (a.k.a. holes) or occluded areas of an image with a natural looking content, image fusion contracts information retrieved from the same scene from similar or multiple sensors and image stitching indicates the composition of multiple images (e.g. generating panoramic views). In the following sections, all methods are related to the morphing approach of this thesis, since an image registration process is necessary. Image registration means that correspondences in different images are identified. Finding correspondences can be either done in a manual, semi-automatic or automatic way. As a result of the morphing process as well as all other methods new visual information is generated leading to altered or even new insights.
The generation of new insights by summarizing visual data is also applied in the research field

of computer graphics. 3 Dimensional (3D) models can be morphed by applying warping and blending computations as described in [17, 47]. Furthermore, Carranza et al. in [12] show a free-viewpoint rendering approach of human motions by combining multiple video views and Finistauri and Xi present in [18] an aircraft wing morphing approach in order to simulate optimal flight performances of an aircraft. Since the content of [12, 17, 18, 47] is beyond the scope, in Sections 2.1.1 to 2.1.6 a detailed description is only given of the mentioned computer vision algorithms.

2.1.1 Image Morphing

The process of synthetically producing one or more intermediate frames which illustrate blended parts of two or more source images is usually known as *image morphing* which is derived from *image metamorphosis* [5]. Due to compelling research results which will be discussed in the following sections, image morphing is applied in the commercial market as well. Some of the first movies which were using these special effects are *Willow* and *Indiana Jones and the Last Crusade* [62].

Traditional methods described in [5], [35] or [62] propose image morphing as a method having two steps, one step for warping, which applies 2D geometric transformation to align features of the source images and one for blending, meaning the interpolation of colors. This can be done to, e.g., merge two faces in a visually aesthetically way. Approaches which are able to handle the morphing task with two or more source images are described in [36] and [29]. The following paragraphs are explaining 2D image morphing in detail and are grouped by the number of source images.

Morphing of Two Images

In [5] Beier and Neely explain an image morphing procedure for computing intermediate frames which illustrate a smooth transition from the first source image to the second. According to [5] the main concept behind image morphing using two images is as follows:

- Find correspondences in two source images S_1 and S_2.
- Based on these correspondences geometrical transformations are applied to S_1 and S_2 which is called warping.
- Corresponding pixel intensity values of S_1 and S_2 are controlled by assigning intensity weights. This step is called image blending. Considering, e.g., the „image in the middle“ the weights of the respective pixels have to be 0.5.

This thesis deals with algorithms finding pixel correspondences in a fully automatic way. In contrast, in early times of developing morphing algorithms exact correspondences are found in a manual or semi-automatic way. The determination of correspondences has to be carried out carefully because incorrect correspondences or even missing correspondences lead to a lack of alignment a.k.a. ghosting [5]. Figure 2.1 depicts a case of ghosting caused by a weighted blending without warping the images before. Parts of the dogs, like tail or snout are not on top of each

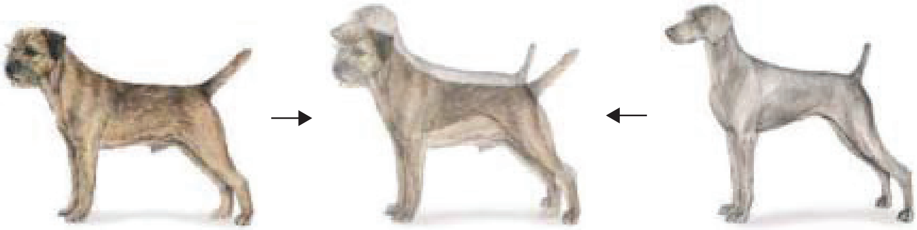


Figure 2.1: The images of two dogs without doing the warping step before blending. Since the features of the two source images are not aligned the resulting image contains regions where ghosting artifacts are visible (figure taken from [49]).

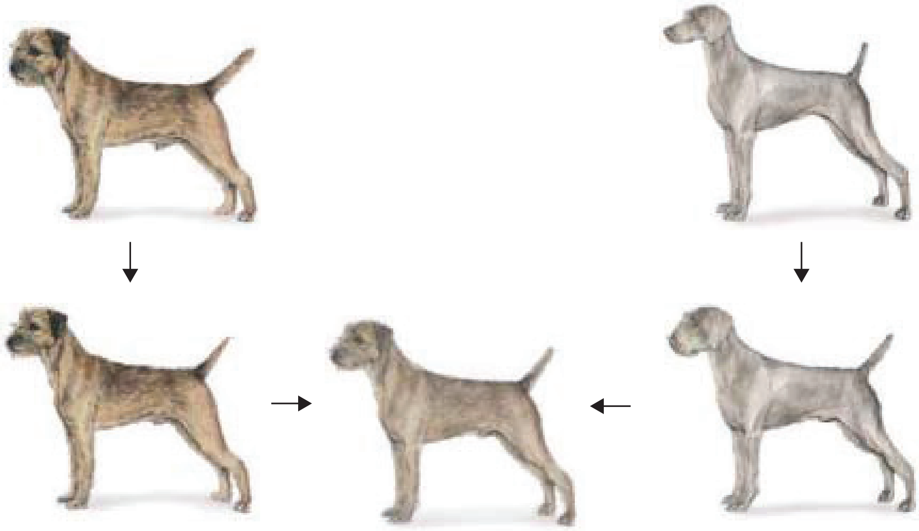
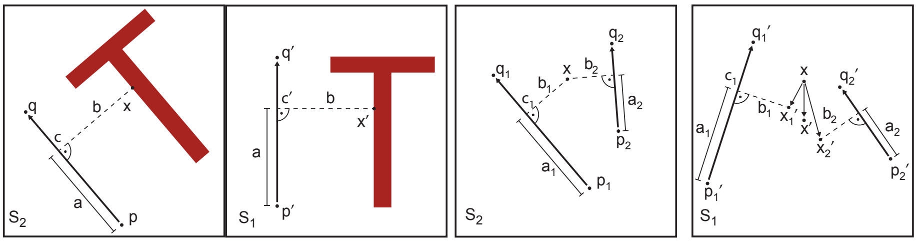


Figure 2.2: The initial images of two dogs (in the top-left and top-right corner) are warped before the blending step (in the lower left and lower right corner). The result shows aligned features in the images. Additionally, no ghosting artifacts are visible in the morphed image (figure taken from [49]).

other. The difference to a morphed image where all parts of the dogs are aligned is illustrated in Figure 2.2. Manually determined pixel correspondences are used e.g. by Beier and Neely who describe a feature-based approach in [5]. They distinguish between two methods of warping where line segments are considered as features:
The transformation builds on the principle that if two corresponding lines of S_1 and S_2 are known, each corresponding relative position to the line can be determined. An example is given where for position **x** in S_2 a corresponding position **x'** in S_1 should be determined. A transformation with one pair of lines means that line **pq** in S_2 has a corresponding line **p'q'** in S_1. A straight perpendicular line to **pq** which crosses **x** produces the intersection point **c**. Furthermore, vector a represents the distance from **p** to **c** and b from **c** to **x**. The two resulting vectors are used



(a) (b) (c) (d)

Figure 2.3: Rotation of the letter „T" with one pair of lines: (a) source image S_2 where line **pq** is manually determined. The intersection point **c** is computed by finding the perpendicular line to **pq** which crosses **x**. Consequently, a is the length from **p** to **c** and b denotes the distance from **c** to **x**. (b) **x'** in the source image S_1 is determined by using the computed values a and b from S_2. Transformation with multiple pairs of lines: (c) destination image where multiple line segments $\mathbf{p_i q_i}$ are determined. (d) Point **x'** is computed by taking u_i and v_i from S_2 and by interpolating the distance from **x** to $\mathbf{x_i}$**'** (adapted from [5]).

to determine **x'** in S_1 [5]. For a better understanding, a rotation of the letter „T" is illustrated in Figures 2.3a and 2.3b, respectively. Applying not only warps which include simple transformations like scale, rotation or translation a more complex warp method is proposed.
When using complex transformations e.g. non-affine transformations on images, multiple line segments $\mathbf{p_i q_i}$ need to be determined in image S_2 which have their corresponding lines in S_1. The first step consists of computing u_i and v_i of S_2 for each line segment in relation to the given point **x**. Based on v_i and u_i for each line segment $\mathbf{p_i}$**'**$\mathbf{q_i}$**'** in S_1, $\mathbf{x_i}$**'** is calculated. The resulting point **x'** is a weighted average of all displacements from **x** to $\mathbf{x_i}$**'**. The idea of using multiple lines is to have a higher influence on points lying nearer to a line. Conversely follows, the greater the distance between line and point the less the influence on the point's shift. The intention of assigning a weight is to have additional visual control of the resulting image [5]. Using multiple pair of lines is depicted in Figures 2.3c and 2.3d, respectively.
A smooth transition between two source images S_1 and S_2 can be realized by interpolating the center position and orientation of each line. Subsequently, corresponding frames are blended by assigning a weight to the pixel's intensity values. The result is a morphed in-between image T.
An approach for generating a morphing transition comprising a semi-automatic image registration process is explained in [35]. Lee et al. show the generation of warping functions by using a Free Form Deformation (FFD). The tedious work of extracting features manually is supported by the snake algorithm [35]. Therefore, it is sufficient to give a position in a coarse manner around the feature. For generating the warping function the FFD requires to put a lattice over the object which has to be distorted and drag the control points of the lattice. The deformation is calculated by using a 2D uniform cubic B-spline surface where the lattice acts as parameter space.
As it is realized in this thesis, Darabi et al. [15] and Shechtman et al. [54] present approaches which execute the registration process in a fully automatic way. These similar approaches are able to morph different textures and structures. Darabi et al. propose an image morphing ap-

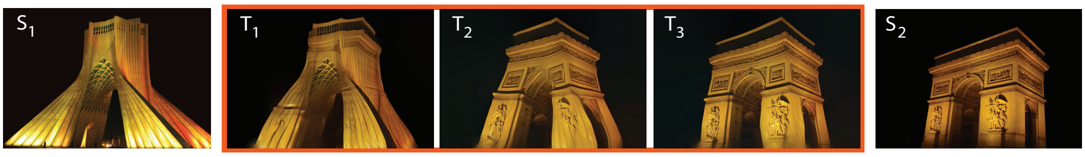


Figure 2.4: A fully morphed transition between S_1 and S_2 with intermediate frames $T_1 - T_3$ implemented with the Generalized PatchMatch algorithm (figure taken from [15]).

proach which is called image melding and uses similar approaches for multiple image editing tasks. The simplest one fills missing pixels of a hole by comparing neighboring pixels. Let T be the target region (hole) and S the source region (neighboring area), Darabi et al. [15] use the Generalized PatchMatch algorithm [4] to find best matching image patches for T from S. This algorithm is described in Section 3.1.1. It helps to minimize an energy function $E(S, T)$ which compares patch distances from T to S. Each patch consists of three color channels and additionally two gradient channels per pixel which is a significant difference to [54]. The gradient channels are weighted and as distance metric the Sum of Squared Distances (SSD) is used. In T best matching patches to S are overlapping [15]. Hence, blending the values of these patches is necessary and called voting. The searching of nearest neighbor patches and the voting step are alternated starting at the coarsest level of a Gaussian pyramid.
Based on the hole filling approach, the image morphing task can be expressed as an optimization of:

$$\begin{aligned} E_{morph}(T^{1\ldots K}, S_1, S_2) = \sum_{k=1}^{K} \{ & \mu_1 E_{BDS}(T_k, S_1) + \mu_2 E_{BDS}(T_k, S_2) + \\ & \mu_3 E_{BDS}(T_k, T_{k-1}) + \mu_4 E_{BDS}(T_k, T_{k+1}) \}, \end{aligned} \quad (2.1)$$

In Equation 2.1 T denotes the intermediate frames of S_1 and S_2 where K is the number of in-between frames. Each in-between frame has to be similar to S_1 and S_2 and to its neighboring in-between frames. To ensure similarity in the images of the transition the Bidirectional Similarity (BDS), described in Section 2.1.6, is used. Therefore, BDS minimizes the energy $E(S, T)$ by alternating S and T as input images which can be described as $E_{BDS}(S, T) = E(S, T) + E(T, S)$. Hence completeness and similarity are given in T. The achievement of completeness and similarity is described in Section 2.1.6. The gradients and colors of T_k are blended by using the given weights, denoted as μ in the optimization term. For smoothing and further optimizing T_k, the Poisson equation, which is explained in Section 3.1.2, is solved [15]. An example of a resulting smooth transition comprising three intermediate frames is illustrated in Figure 2.4.

The content of [17, 33, 47, 55, 66] is beyond the scope of this thesis but it should point out that image morphing is executable in a 3D image space as well. In [66] Xu et al. propose a method for shape interpolation with the support of Poisson editing. By linear interpolation new vertices are computed to form a triangular mesh. Further 3D morphing approaches are described in [17, 33, 47, 55]. 3D morphing can be applied in industrial design research areas where new product shapes are obtained from existing ones or in computer animation and computer graphics [33].

Morphing of Multiple Images

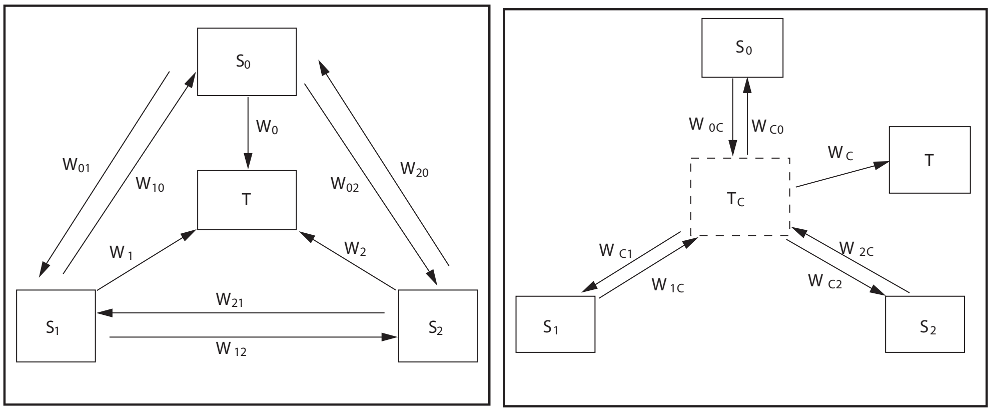
As this thesis pursues the objective of extending the range of source images to more than two, this section presents morphing approaches coping with at least three source images. This can be useful for completion tasks or image collages, where regions of several images are selected and merged to one seamless image.

Lee et al. use the multilevel free-form algorithm, described in [34] to generate warping functions from the manually arranged correspondences in the images. For minimizing the work of an animator to mark features for $k(k-1)/2$, image pairs warping functions are propagated. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected Graph, where $\mathcal{V} = (v_1, v_2, ..., v_K)$ denotes K (the number of source images) vertices and $\mathcal{E}$ the connecting edges in-between. An edge from v_k to v_j exists, if a warping function W_{kj} is already derived. The initial graph is connected and owns $(k-1)$ edges. W_{ij} can be propagated if a vertex v_k is found where edges from v_k to v_i and from v_k to v_j exist.

In [36] Lee et al. present a morphing method called polymorph. Firstly, correspondences are computed semi-automatically by using thin-plate splines and multi-level free-form deformations. For each pair of source images S_i, a warping function W_{ij} exists which maps each point in S_i to its corresponding one in S_j. If K denotes the number of source images each node of a $(K-1)$ - dimensional simplex represents a source image and each edge a warping function, as illustrated in Figure 2.5a. If all points in this simplex are given in barycentric coordinates a blending vector determines the position of the intermediate image and thus the influence of S_i on the finally morphed image. To compute a target image T, W_i is derived by linearly interpolating W_{ij}. By linearly interpolate the resulting images T_i the image's color values leads to the target image T [36]. Blending T_i by treating each pixel value equally (uniform) is not enough if only few regions of each T_i are selected. This case is illustrated in Figure 2.6 where hair, nose and eyes are selected from different source images. Thus, a blending function B_i is introduced containing weights for each pixel value in order to achieve an individual treatment of every pixel (nonuniform). Due to a significant memory overhead when computing K^2 in-between frames the framework is optimized by establishing a central image T_C, where for each source image two warping functions W_{iC} and W_{Ci}, are calculated. The introduced blending vector B_C determines the relative influence from S_i on T at a given position and has to be established by the user [36]. Selected regions R_i of S_i are mapped on T_C by using the warping function W_{iC}. By linearly interpolating W_{Ci} with the weights of B_C the warping function W_C is derived. Figure 2.5b depicts the morphing approach with the optimization of adding a central image and a blending vector determining for a selected region the influence of a source image.

In order to avoid unnatural looking in-between images pre-processing and postprocessing methods are applied to S_i and T. The pre-processing step contains distortions of feature shapes and positions for an enhanced alignment, whereas the postprocessing step is used for local refinements and global manipulations [36].

Finally, an approach where correspondences are found automatically is presented in [29]. Jones et al. introduce a technique to represent an image-class of objects with the help of a multidimensional morphable model to finally classify novel images by minimizing an error function. A reference image of a data set containing one class of objects is determined. For each prototype image (= source image) pixel-wise correspondences to the reference image are computed auto-



(a) (b)

Figure 2.5: (a) Target image T is computed by firstly finding warping functions W_{ij} for each pair of images, which are afterwards linearly interpolated for each i to get W_i. Applying W_i on S_i leads to the distorted image T_i. Finally, all T_i's color values are interpolated (specified by the blending function B_C) to get T. (b) The optimized Framework provides a central image T_C to save storage. T_C is placed in the centroid of the simplex and the warping functions W_{ij} are not saved anymore. Instead the warping function to the central image W_{Ci} / W_{iC} is determined. The warping function W_C (from T_C to the target image T) is calculated by linearly interpolating W_{Ci} with the weights of the blending function B_C (adapted from [36]).

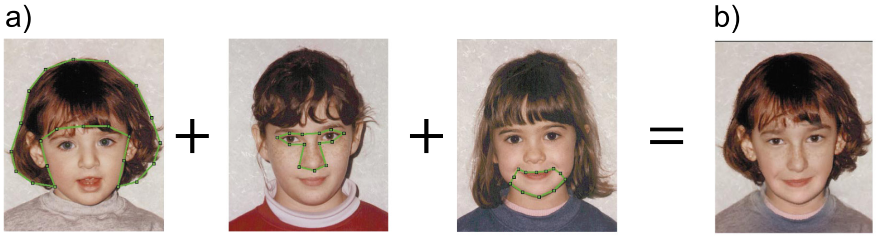


Figure 2.6: (a) A result of the polymorph algorithm where three source images are used. Manually marked image parts (within the green border) are selected for the computation of (b) (figures taken from [36]).

matically with the help of a bootstrap algorithm, described in [60]. An image is represented by a shape vector determining the relative displacement of each pixel to its source pixel of the reference image and a texture vector containing the difference of pixel intensities to corresponding positions. Once the warping process is done the shape of the prototype image matches the one of the reference image. Each image which gets warped to a reference is added to a pool where correspondences from every image to each other exist. As a consequence, this approach enables to represent images of one class by a linear combination of vectorized prototype images [29].

2.1.2 View Morphing

The term view morphing is used in this work to describe the computation of a new viewpoint for a virtual camera focusing an object. The viewpoint is obtained by interpolating the data of two source images illustrating the given object from different viewpoints. View morphing can be seen as a subtopic of image morphing concentrating on morphing different perspectives of a scene. In contrast, traditional image morphing as it is used in this thesis is for instance not considering different viewpoints of a scene or any extractions of geometric peculiarities. However, view and traditional morphing presented in this work have in common that a registration process of the source images is necessary. Especially Pérez et al. describe the image registration process (finding correspondences) as a challenging task of the algorithm. In their work view morphing is defined as a shape-preserving extension of image morphing [53]. To prevent ghosting and shape distortions in the transition process image registration is accompanied by manual inputs. Due to the necessity of establishing exact correspondences the influence of an animator is required in view morphing approaches, such as [42,53,65]. Algorithms which compute correspondences automatically can use additional informations like position and orientation from calibrated cameras [13]. For an interpolation of the object's shape either projection matrices are computed to realize object transformations or e.g. common morphing tasks (as presented in [5,53]) are used. Chen and Williams describe in [13] their interpolation as *„an approximation to the transformation of the pixel coordinates by a perspective viewing matrix"*. Since background and foreground have different epipolar geometries they are morphed separately. Splitting into two layers is performed manually in [42,64,65]. Chen and Williams refer to overlapping areas caused by images containing common scenes as a potential source of error during the morphing process which is prevented by using a Z-buffer algorithm to maintain the pixel nearest to the lens.

View morphing results are applied in applications like virtual holograms or walkthroughs in virtual environments, image-based primitives or incremental rendering [13]. A benefit of this method is that the knowledge about 3D shapes is not required and thus computing in-between images is independent of the scene complexity [13,42,51,53,64].

2.1.3 Image Fusion

Having two or more single images which contain informations retrieved from similar or multiple sensors, the merging of the images is commonly called image fusion. When multisensor data is fused the scene of the image remains the same which is a considerable difference to traditional image morphing methods like [54] in which the morphing of different objects or scenes is focused. However, both methods have to register the source images. A resulting fused image enhances the human or mechanic „readability" of the image [1]. The measurement of an image's information content is indicated as entropy rate [1]. The maximum information can be reached if each of the images' gray levels have the same frequency [1]. Consequently, the higher the information content of the fused image the higher the entropy rate. Similarly to image morphing the method requires an image registration process, which determines correspondences in the images. To obtain a final image, single pixel values have to be selected from possible candidates of source images. Trivial methods are the maximum selection method which selects the pixel value with the highest intensity, the simple average method which averages the pixel values, or the minimum selection method which takes the pixel value with the lowest intensity. Further methods to determine the pixel value for the merged image are the Laplacian pyramid method where Laplacian pyramids of the source images are computed to get the resulting fused image and the Principal Component Analysis (PCA) method which helps to decrease multidimensional data sets to lower dimensions [1]. In the research area of Medical Imaging a common approach is to merge relevant informations obtained from Computer Tomography (CT) and Magnetic Resonance (MR). Furthermore, image fusion can be applied to various types, e.g. multispectral and panchromatic images of satellite images [67] or in the scope of Robotics, where different trackers provide different informations and image fusion serves for a improvement of navigation [41].

2.1.4 Image Stitching

Image stitching is the process of seamlessly merging two images within a common overlapping boundary region. In contrast to image morphing where the whole dimensions of two images get registered to each other stitching requires to find common regions only in overlapping areas. This approach can be used for generating a panoramic view of several single images, virtual reality, super resolution or texture synthesis [52]. Two main problems have to be handled:

- Finding a best common boundary of adjacent images.
- Adapting the color intensities of the images to obtain a seamless transition.

To find the optimal seam, in [52] Sadeghi et al. use an approach where pixel intensity values are compared within overlapping regions. By employing a minimization weight function, which computes the Sum of Absolute Difference (SAD) (l_1-norm) of the pixels, an optimum with the slightest intensity difference is found. To adapt the intensities to each other a color correction based on the Poisson equation, which is described in Section 3.1.2, is used [52]. In [26] Jia et al. use dynamic programming and graph cut for minimizing intensity and gradient values in two images to find the optimal seam. The method of Jia et al. [27] is robust against motion blur and occlusions by using a tensor voting method. According to investigations of Jia et al. [28] it is not enough to produce a seamless transition and subsequently adapt the intensities of the images. There are two more requirements which have to be followed in an image stitching process. Firstly, the computed seam must not break through a salient object since disturbing artifacts (ghosting) could be produced. Second the form of an object should stay consistent and not be deformed by attaching a foreign object of the other image (considering a tree trunk which appears thicker because of an incorrect computed boundary). The proposed algorithm in [28] determines the quality of the computed features in the overlapping area. For this measurement either the Single Optimal Partitions (SOP) or the Double Optimal Partitions (DOP) technique is applied. Along the computed partitions 1 Dimensional (1D) features are computed to which deformation vectors are associated for the matching process. Finally, the deformation vectors are propagated toward all other pixels.

2.1.5 Image Completion

Image completion is a image editing method to fill unknown or occluded areas of an image with a natural looking content. Applying image completion on an image requires only one source image. This means that the content filling occluded or unknown areas has to be found in neighbored regions of the affected area. In state-of-the-art image completion methods missing informations of a hole are substituted by surrounding nearest neighbor patches. In [15] Darabi et al. describe the image completion task as a minimization of the SSD of two related patches. For initialization pixel values at the hole boundary are interpolated. Establishing a Gaussian pyramid from coarse to fine patches of the hole are substituted with nearest neighbor patches from known regions with the help of the PatchMatch algorithm (described in Section 3.1.1). Optimizations are done by additionally considering the orientation of given patches and solving the discrete Poisson equation, described in Section 3.1.2. In [61] Wexler et al. present a similar image completion method for images and video editing. Possible applications are the elimination of unwanted objects in video sections, the visual enhancement of old damaged video frames or modifying an undesired behavior of an actor in a video sequence. Based on the algorithm of [61] Kopf et al. propose an application where the quality of the upcoming completion task can be predicted to select a region that can be successfully completed [32].

2.1.6 Bi-Directional Similarity

Simakov et al. describe in [56] a method for summarizing visual information which can be used for image scaling or the summarization of videos in a temporal manner. Image scaling explains either the process of decreasing (re-targeting) or increasing (image synthesis) the image's size. To concentrate on images the main goal for image re-targeting is on the one hand to preserve as much information of the input data (e.g. objects like buildings, trees, persons, etc.) as possible (completeness) and on the other hand to render a „naturally looking“ image (coherent) which avoids incomplete rendering of objects (considering a building where roof and windows are missing). BDS is used in the morphing process presented in [15]. Considering a morphed transition from one source image to the other then BDS ensures for each in-between frame the completeness and coherence to its consecutive frames.

These constraints are reached by firstly defining patches in both the source image S and target image T. The completeness term states that T should contain as much patches as possible from S. In contrast the coherent term states that as much patches as possible from T should be contained in S. If R denotes a patch of the source image S and Q a patch of the target image T then Q is a corresponding patch or nearest neighbor of R if the SSD between Q and R is at a minimum and vice versa. Patches are defined for each pixel and in a Gaussian pyramid for each level. The Gaussian pyramid contains multiple scales of the images and is introduced to capture a bi-directional similarity locally and globally. For initialization a good initial guess of the target image T is required. The better the initial guess of target image T the better the results of the bi-directional similarity algorithm [56]. It would be a bad guess if T is simply cropped, since it may happen that important patches of S are lost. If T is smaller than 95 percent of S then resizing T is a bad option as well [56]. Thus, a gradual resizing process is implemented. Intermediate frames are produced by decreasing the size of S. For each in-between frame and its source image the bi-directional similarity measure and relevant refinements are executed iteratively.

2.2 Numismatics

Since this thesis is specialized in morphing ancient coin images this section provides background information about the coin's genesis starting at the manufacturing up to computer vision applications on ancient coin images. According to [21] the term numismatics is not uniform throughout different kind of scholars. The term originates from the sixteenth and seventeenth centuries where collectors tried arranging and acquiring knowledge about certain kind of pieces which they called medals. Each small object consisting of metal was countered as medal but no account was taken about the function of these medals but rather the design and outer appearance. Scholars of the later centuries extended the domain of definition for numismatics with the result that all portable objects which might be used as a mean of exchange are numbered among due to the fact that the main function of medals from former centuries was economic [21].

2.2.1 Making of Ancient Coins

The production of ancient coins was realized either by striking or casting. The earliest bronze coins from the Roman Republicans were e.g. cast because of their large diameter [21].
The making of dies was executed by specially qualified moneyers. A mint consisted of an obverse die, a blank and a reverse die. The designs were cut freehand (die-cutting) and fixed at the upper face of the anvil and the lower face of the punch. With iterative blows of a hammer on the upper side of the punch the blank got struck simultaneously its head and tail. Figure 2.7 depicts a mint where blanks become coins. Due to the manual production of the coins, problems could arise in the process of striking. The blank might have cracked under the pressure of the hammer blow. Coins could get pierced producing a hole in the coin or an existent coin was used as a blank leading to possible residuals of the old design in the newly struck coin. Every hand-made coin die was individual in its appearance and had to be renewed due to wear marks after several iterations of minting (about 10.000 iterations) [21]. The dies of obverse and reverse die might wear off in varying rates which means that the dies are not only substituted as a pair

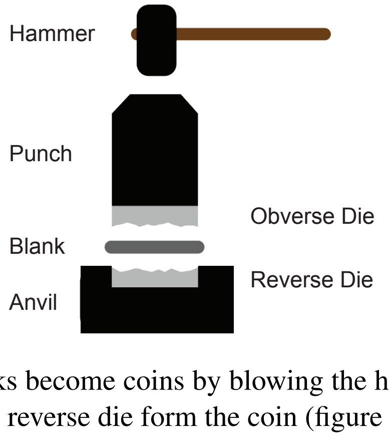


Figure 2.7: A mint where blanks become coins by blowing the hammer on the upper end of the punch. Finally, obverse die and reverse die form the coin (figure taken from [74]).

but rather be replaced individually. Thus, coins are struck by various combinations of obverse and reverse die. In order to still determine coins of common types numismatics is investigating coin linkage, which is described in [16].

2.2.2 Numismatics in Combination with Computer Vision

The research area of computer vision tries to interpret information from image contents. Arranging and acquiring knowledge about ancient coins is a field of research on numismatics. Initially, to connect numismatics with computer vision, the coins have to be photographed and converted to digital images. The digitization process of coins requires to ensure that as less as possible of the visual information gets lost. To get an understanding concerning the way digital high-quality images are produced from coins Goodman guides in his work [20] through several steps of coin photography. Once coin images exist several tasks in the field of computer vision have emerged. Potential research fields for the future are given in [74]. For this work only existing applications are mentioned.

Coin Classification

The problem of coin classification is to categorize coins according to a predefined criterion. A criterion could be e.g. the type or the reference number(s) of a coin, which can be taken from reference numismatics literature (e.g. [14]). The coin is classified correctly if the assigned criterion equals the ground truth. Doing coin classification automatically on digital coin images shows to be a challenging task. One reason is the large number of given coins (e.g. 550 types for the Roman Republican period [14]). Furthermore, on the one hand the difference between coin types is low and on the other hand the intra-class variability (which indicates the difference of the visual appearance within one class) is high [75]. In Figure 2.8 an example is given where coin types are different but simultaneously highlighting a high intra-class variability. Van der Maaten [59] describes that differences exist between classification tasks of present and ancient coinages. In [69] Zaharieva and Kampel explain that for classification systems of modern coins a rotational symmetry of coins, which means that their diameter is known, is assumed. This pre-condition distinguishes them from classification systems of ancient coins. In contrast, the shape (and thus the diameter) of ancient coins appears almost arbitrary which makes the task more complex. Classification methods based on data sets of modern coins are given in [23, 45, 58]. One of the first encouraging approaches attempting to recognize ancient coins is presented in [68] which makes use of local SIFT descriptors [9]. An average classification rate of 84.24% is obtained by using a small data set of 350 images with three different types of coins. The focus of [31] lies on new combinations and extensions of local image descriptors obtained from located interest points of ancient coins in order to ensure a basis for the classification task. The recognition of coins is then performed by feature matching. With the usage of Shape Context [7] combined with a Hessian Laplace detector the best classification rate is located at 92.57% and uses the same image data set as [68]. In [71] Zambanini and Kampel use the energy resulting from the SIFT flow approach in order to determine the image similarity of ancient coins on a Roman Republican coin data set with 24 classes where each one comprises three images. As a result 74% of the coins are classified correctly. With a common evaluation procedure the method outperforms [31]

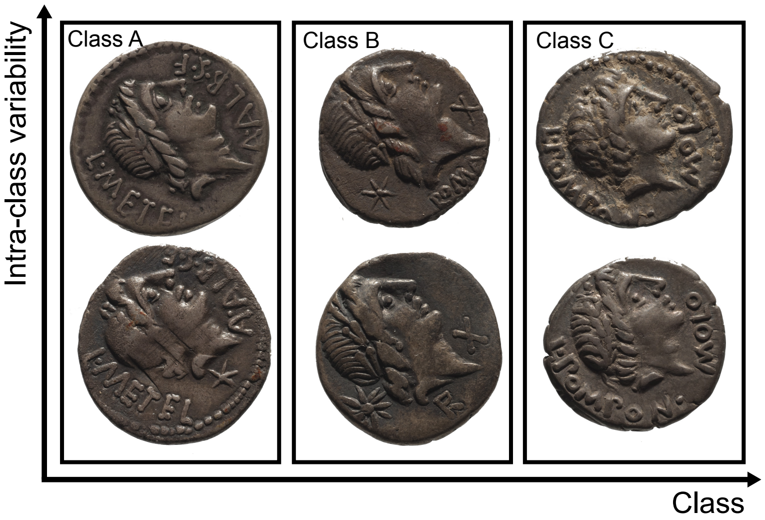


Figure 2.8: A selection of Roman Republican coins showing a high intra-class variability (columns) and a high similarity of the coins among the classes (rows).

with more than 40% where both are applied on the same (and new) data set. An extension of the SIFT flow method is proposed in [72]. The data set for this evaluation comprises 60 classes and achieves a result of 83.3% of correctly classified Roman Republican coins. Besides, a further improvement lies in the performance of the classification task by introducing a coarse-to-fine classification procedure. Furthermore, the classification rate in [75] passes the 90% mark by combining exemplar-based and lexicon-based legend recognition. In [2] Anwar et al. use the Bag Of Visual Words (BoW) technique to classify ancient coins. They establish a dense grid to extract SIFT descriptors and match them against a vocabulary by using the Eucledian distance. Each descriptor represents a visual word which is then registered in a histogram. Considering the spatial location of the visual words as well, rectangular tiling, log-polar tiling and circular tiling are employed. In the part of evaluation the circular tiling outperformed all other methods due to the rotational invariance. The published work of Quraishi et al. presents a classification method of ancient coins with the help of a neural network. A classification rate of 75% is reached on a data set with only 20 images [50].

Further Applications

To get a better idea concerning the way computer vision interacts with the research field of numismatics further applications are explained which have emerged in the past ten years:
As a pre-processing step to computer vision tasks performed on ancient coin images (e.g. coin classification or coin identification) an image has to be segmented into foreground regions representing the coin and background regions. Segmentation tasks are proposed in [70] where [70] can be executed fully automatically thus providing the benefit of handling a large variety of coin image styles. The identification of a numismatic object requires to localize unique features which enable the differentiation of the object from all other ones belonging to the same class.

Interesting features can be computed e.g. by considering scratches, wearing, shape or varying production processes [30]. The shape of an ancient coin is not exactly circular which means that it can be a helpful attribute for the identification process. Deviation from circular shape matching (DCSM) is the name of a method presented in [24] where the algorithm is based on the description of a coin's shape border as the deviation from a circular shape. DCSM measures the polar distance from the sampled points of a coin's shape border to a circle and stores it in a vector. The result is a 1D descriptor representing the border of a coin. With this application Huber-Moerk et al. reach an identification rate of 95.16% [24]. An extension of this approach is given in [25] where the identification rate is increased to 99%. Furthermore, an application is demonstrated in [76] where Zambanini et al. use a state-of-the-art scanning device to generate 3D models of ancient coins. The acquisition of 3D models from ancient coins can be useful e.g. to examine for changes on the coin's surface like cuts or to get a better insight to the coin's features by simply changing the viewpoint [76]. Mudge et al. acquire 3D geometries from 2D images containing ancient coins by applying Polynomial Texture Mapping (PTM). With the help of a low cost system fifty identically aligned images with different illuminations per image are taken to achieve the PTM light direction images and the 3D geometry. The usage of this procedure leads to more accurate 3D virtual coins and enables to create a 'virtual exhibition' [44].

2.3 Summary & Implications for the Proposed Methodology

Summarization methods provide a possibility to bring given images' content into relationship with each other by applying an image registration process. While the registration process is part of all presented methods the output is different. Image morphing blends multiple images to a single one, the view morphing produces a synthetic viewpoint from two images with different views of a scene, image completion fills unknown or occluded areas of an image with a natural looking content, image fusion contracts information retrieved from similar or multiple sensors of the same scene and image stitching treats the composition of multiple images (e.g. to generate panoramic views). For this thesis, the SIFT flow algorithm is used in order to register images due to the high flexibility of the algorithm: the SIFT features used are invariant to scale and partially invariant to changes of illumination and 3D viewpoints [39]. Moreover, a dense correspondence search is done which enables to warp/align images [37] and a classification of ancient coins can be realized by exploiting the energy term as similarity metric which is presented in [72]. The extraction of image patches during the morphing process is inspired by the PatchMatch algorithm, however, the idea of selecting more than two source images for the morphing process is transferred into this work. The term numismatics originates from the attempt of collectors to acquire and arrange knowledge about coins which were used as a mean of exchange. Coin classification aims to categorize coins according to a predefined criterion like the type of a coin. In contrast, the identification of a coin requires to localize unique features which enables the differentiation of the coin from all the other ones belonging to the same class. Shape matching, which presupposes the segmentation in foreground and background, represents a possibility for identifying coins. Furthermore, the acquisition of 3D models from ancient coins helps to get a better insight to the coins' features by means of a 'a virtual exhibition'.

CHAPTER 3

Methodology

In this chapter a new approach for image morphing is presented. The theoretical background of already existing algorithms is expounded in Section 3.1. Based on these foundations, a system is described which morphs images originating from a common class in order to receive a class representation. The system handles two or more source images and delivers as output one image containing a visual summarization of similar image regions. A detailed explanation of this work's core element, the multi-image morphing framework, is given in Section 3.2.

3.1 Background

For a better understanding, this section comprises a theoretical background to the multi-image morphing approach. The approaches presented in the following are used to register images to each other and subsequently execute optimization operations on images.

3.1.1 Image Registration

Zitová and Flusser describe image registration as *„a process of overlaying two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors“* [77]. Several years after the publication of [77] two image registration algorithms were developed. Firstly, in 2004 SIFT [39] was introduced and six years later Barnes et al. [4] presented the Generalized PatchMatch algorithm. Both allow a registration process of scenes deviating slightly from each other.
At this point image registration can be described more generally as the process of finding correspondences in related image contents. Correspondences might be found as matching points, lines or segments. Optical flow methods are developed to estimate the transitional change of the image's content by comparing pixel intensities [38]. In other words a dense sampling in time is done which enables tracking of a moving object. By comparing the images a movement vector is assigned to each pixel of the scene. Optical flow techniques can be divided into local and global methods whereby the first category comprises the approach of Lucas and Kanade [40]

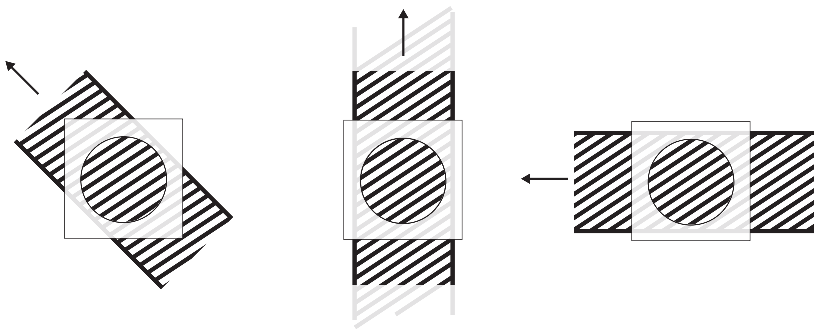


Figure 3.1: Through the circular window a movement of the bars is perceived but not the exact direction. Possible movement directions of the whole object behind the window are up-left, left or up leading to the same result namely the movements of the bars from bottom right to top left.

and the latter one Horn and Schunk's method [22]. A possibility for combining a global and a local method is presented in [10]. Local methods provide higher robustness under noise than global do, although global approaches return a dense flow field [10]. A problem which can not be solved by optical flow estimates is the aperture problem. Considering a small window through which movements of an object can be perceived, the direction of the object's movement can not precisely be determined. Figure 3.1 illustrates the problem where the exact movement direction of the whole object can not be determined when only considering the circular window. SIFT flow is based on optical flow methods but uses SIFT features and provides dense sampling in the space of natural images which enables the alignment of images [38]. The Generalized PatchMatch algorithm also aligns images but different to SIFT flow by comparing image regions called patches [4].

Scale Invariant Feature Transform

This section explains the SIFT algorithm since this forms the basis of SIFT flow which is explained in Section 3.1.1. SIFT is able to extract distinctive invariant features of images [39]. The extracted features are invariant to image rotation, scale and partially invariant to changes of illumination and 3D viewpoints. Due to its robustness the features are usable for matching similar scenes in two images. The computation of SIFT features can be divided into four steps which are explained in the following.

Scale-space extrema detection The first step intends to find potential pixel candidates which can be recovered independently to scale and orientation of differing scenes. To find scale invariant points it is necessary to search for these features in all possible scales of the given image. Therefore, a Gaussian function G serves as kernel and convolving it with the given image I leads to a scale space. An octave of the scale space consists of Gaussian images, resulting from the convolution of G and I, and the Difference-of-Gaussian (DOG) which results from the difference of adjacent Gaussian images. Figure 3.2 illustrates the first and second octave of a scale

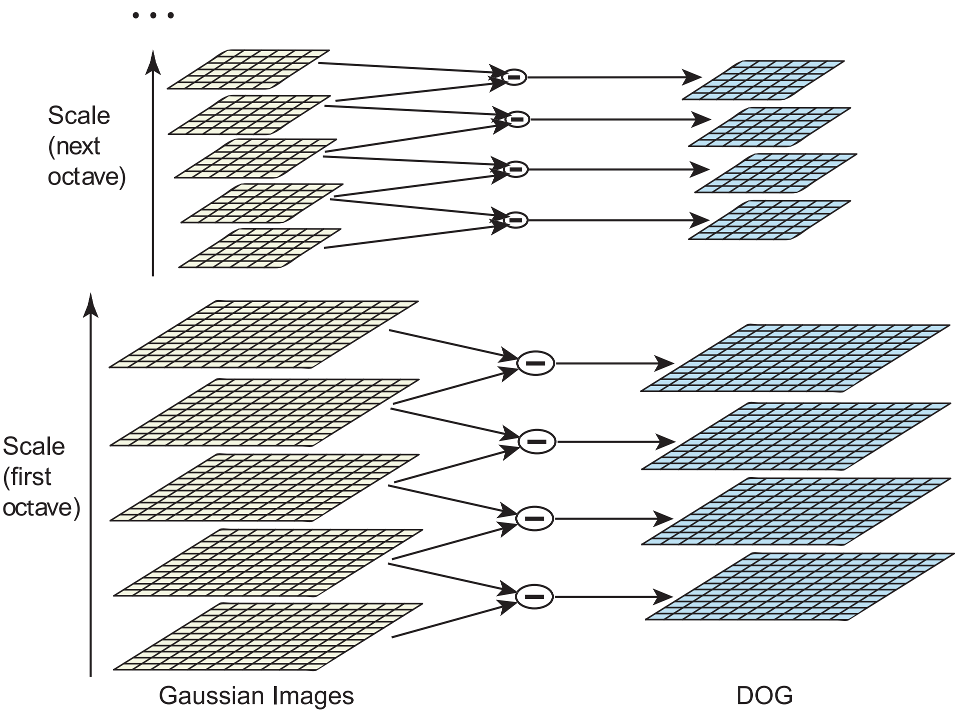


Figure 3.2: A scale space for detecting scale invariant features consists of octaves where each one is composed of Gaussian images (left) and the DOG (right). Gaussian images result from the convolution of a Gaussian kernel and the source image. The difference in two adjacent Gaussian images leads to a DOG image. For each following octave, the size of the image is divided by 2 (figure taken from [39]).

space. The size of the Gaussian images remains the same within one octave and is reduced afterwards by the factor of 2. Furthermore, only the DOG images are considered to spot local minima and maxima. Figure 3.3a depicts the selection of the pixel neighborhood. Each pixel in the image is compared with its eight-neighborhood in the same level and with each adjacent pixel of the 3x3 matrix in the level above and below. If the value of the selected pixel is the lowest or highest of all compared pixels it is marked as a local minima or maxima, respectively.

Keypoint localization Due to inaccuracies the initial detection of a pixel position as maxima or minima is improved. With the help of the 3D quadratic Taylor expansion an interpolated location of the extrema is found. The interpolation is performed using the sample point as origin. If the computed offset from the point is larger than 0.5, the sample point changes and gets the new origin of the interpolation.

Not all of the gained keypoints are stable enough to be used for further approaches. Firstly, extrema with low contrast are eliminated by using again the Taylor expansion for calculating an intensity value. If the result is below a threshold, the extremum gets rejected.

DOG images may depict keypoints along edges having a weak stability due to small amounts of noise [39]. For eliminating these keypoints a Hessian matrix is used in order to compute a principal curvature [39]. The curvature is large when crossing the edge and small when pointing in the perpendicular direction. Due to the proportionality between eigenvalues and curvatures, eigenvalues can be computed instead. It is even sufficient to determine the ratio of the eigenvalue with the largest magnitude and the smaller one [39]. Consequently, keypoints holding a ratio higher than a certain threshold are eliminated. In metaphorical terms this means that keypoints located on edges and smooth regions are rejected and those located on corners are preserved.

Orientation assignment Rotational invariant features means that keypoints are still matching despite rotating one of the source images. Transforming keypoints into rotational invariant features requires the assignment of an orientation. The Gaussian image which is closest to the keypoint's scale is used for computing the gradient magnitude and orientation around a keypoint. The values are registered in an orientation histogram divided into 36 bins (every bin contains 10 of 360 degrees) where every added sample is weighted by its gradient magnitude and by a Gaussian-weighted circular window [39].

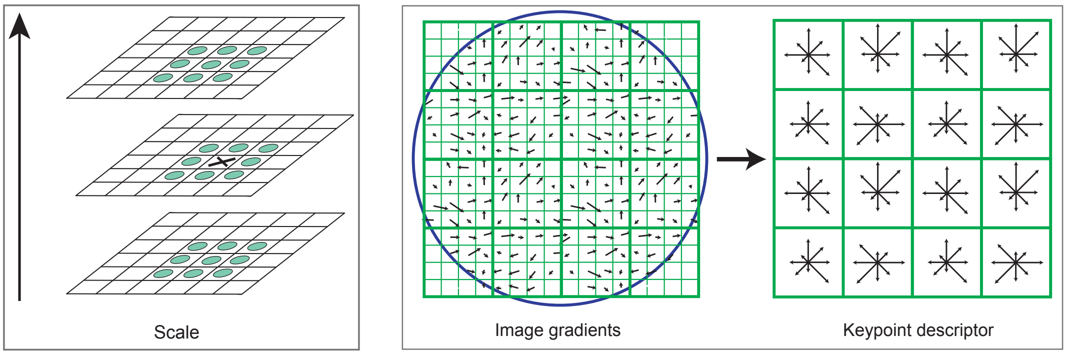
In order to assign the orientation to the keypoint, the highest peak of the histogram indicating the dominant gradient direction and all local peaks, which are within 80% of the highest peak, are selected. In case of finding multiple peaks for every orientation multiple keypoints with the same scale and location are generated [39].

Description of keypoints For all generated keypoints a robust description is necessary providing as much invariance as possible to scale, rotation, illumination, change of 3D viewpoints and small amounts of noise.

Firstly, the used image is blurred by the factor of the keypoint's scale and the gradient magnitudes and directions are computed, as illustrated in the left part of Figure 3.3b. The blue circle represents a Gaussian function with a standard deviation of 1.5 times the descriptor's width and intends to weight the gradients magnitude. The greater the distance from the center the less influence of the magnitude. Furthermore, the usage of a Gaussian function reaches a certain stability against small displacements of the descriptor's window [39]. The right part of Figure 3.3b depicts established orientation histograms over 4×4 sample regions where every gradient is classified by its direction to one out of the 8 bins. Depending on the magnitude and the distance from the center the gradients have more or less influence to the assigned bin. Due to the generation of 8 bins for all 16 (4×4) sample regions the resulting descriptor is a normalized feature vector containing $16 \times 8 = 128$ elements [39].

The descriptor is not invariant to non-linear illumination changes [39]. These changes can effect the length of gradients. In order to reduce this effect, all magnitudes which have a value larger than 0.2 are decreased to 0.2 and the descriptor is again normalized to unit length [39].

Rotation invariance is reached by using relative gradient directions to the keypoint's orientation [39].



(a) (b)

Figure 3.3: (a) All marked pixels (3×3 matrix in current and adjacent levels) in the illustration are compared with the selected pixel for detecting minima and maxima values. (b) Left: computed gradient magnitudes and directions in a squared 16×16 matrix. A Gaussian function (blue circle) weights the magnitudes and ensures a stability of small displacements of the descriptor's window. Right: all gradients of a 4×4 region are accumulated in a 8 bin orientation histogram. The assignment of the 8 bins depends on the gradient direction. Each bin represents one element in the final keypoint descriptor (adapted from [39]).

Scale Invariant Feature Transform Flow

In [38] Liu et al. describe the SIFT flow algorithm which is based on the optical flow principle. In the process of finding SIFT flow correspondences the algorithm uses SIFT descriptors. For every pixel of the source image a descriptor is computed, where the sample region has 4×4 cells and hence leads to a feature vector containing 128 elements. All descriptors of one image form a SIFT image [38]. Once the SIFT images are computed, an energy function is defined, where p is the image position and $\mathbf{w}(p) = (\mathbf{u},\mathbf{v})$ denotes the flow vector consisting of a horizontal and a vertical flow element. Let s_i be the SIFT image then

$$E(\mathbf{w}) = \sum_p min\Big(\big|\big|s_1(p) - s_2(p + \mathbf{w}(p))\big|\big|_1, \omega\Big) + \tag{3.1}$$

$$\sum_p \eta\Big(|\mathbf{u}(p)| + |\mathbf{v}(p)|\Big) + \tag{3.2}$$

$$\sum_{(p,q)\in\epsilon} min\Big(\alpha|\mathbf{u}(p) - \mathbf{u}(q)|, \psi\Big) + min\Big(\alpha|\mathbf{v}(p) - \mathbf{v}(q)|, \psi\Big). \tag{3.3}$$

In the matching process Equation 3.1 penalizes excessive deviation from the flow vector, Equation 3.2 tries to keep the vector small and Equation 3.3 is responsible to keep adjacent pixels' flow vector similar [38]. The variables ω and ψ are denoting a threshold splitting outliers from non-outliers and ϵ stands for all pixels in the 4-connected neighborhood of p.

Any pixel in the source image can possibly match with any pixel in the target image which causes performance problems in the algorithm [38]. Consequently, a SIFT pyramid is established for each SIFT image where for each pixel the flow vectors are estimated in the highest

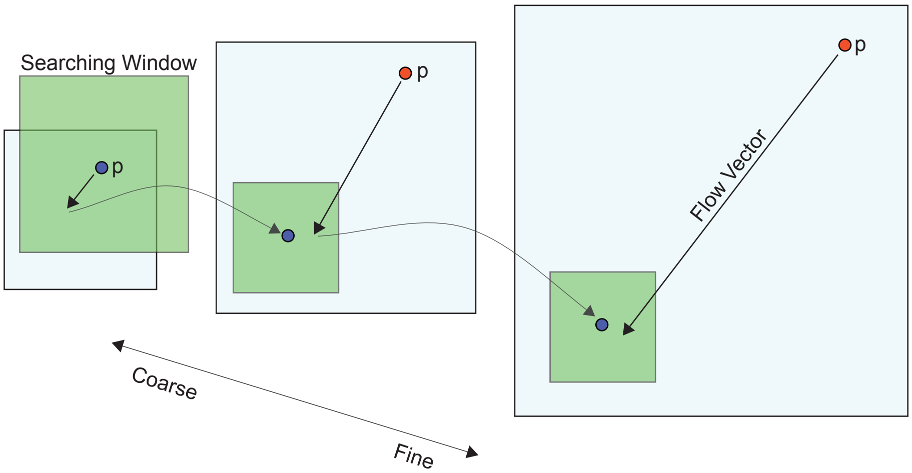


Figure 3.4: The SIFT pyramid generated for coarse-to-fine SIFT flow matching to increase the performance of the algorithm. The searching space is reduced to the size of the green window and decreases the complexity of the algorithm from $O(h^4)$ to $O(h^2 \log h)$. Finding the flow vector with starting point p is done by reducing the energy function composed by Equations 3.1, 3.2 and 3.3 (adapted from [38]).

level (the coarsest resolution of the image). The nearest neighbor pixel has to be found within a searching window. For an easier understanding in Figure 3.4 one SIFT pyramid is illustrated for two SIFT images: one containing a given pixel p and the second a searching window to which the flow vector is pointing. If the flow vector is calculated by minimizing the energy function, subsequently the vector is propagated to the next finer level. This procedure is repeated for each level in the pyramid until the finest one is reached [38]. Due to the features' robustness they are usable for matching similar local structures which, e.g., enables to query a large database and find the nearest neighbor of an image. In other words, SIFT flow performs dense sampling in the space of all images which enables scene alignment [38]. Liu et al. show compelling results by finding similar frame scenes tested on overall 731 videos where each one provides one frame as query image and subsequently the nearest neighbors are found. Figure 3.5 illustrates the results of the query. Figure 3.5a shows the query image and 3.5b the most similar image, determined by the SIFT flow algorithm.

PatchMatch and Generalized PatchMatch

PatchMatch is an algorithm for determining similar patches in two images [3]. The most similar patch compared to a source patch is called nearest-neighbor and is found by a dense approximate correspondence search. PatchMatch is useful for image editing tasks such as texture synthesis and completion, image re-targeting or image reshuffling. In order to measure the distance from one patch to another, SSD is used and in the correspondence search only translations are considered. Thus, Barnes et al. extended the algorithm in [4]. The Generalized PatchMatch algorithm shows clear benefits compared to PatchMatch:

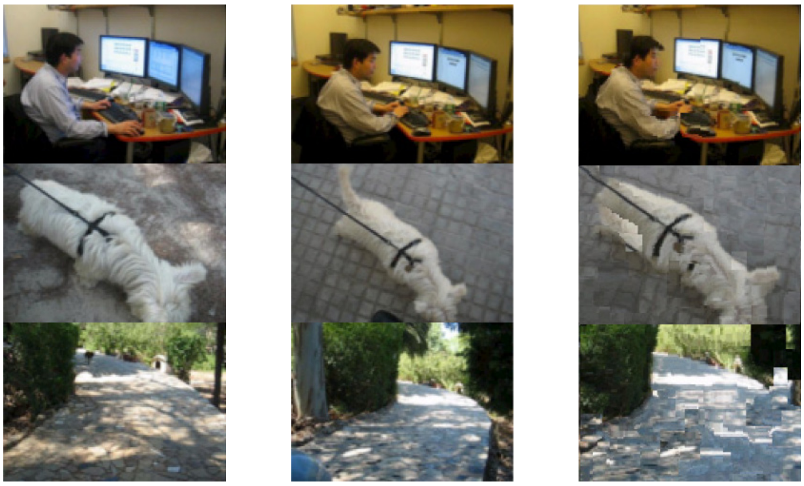

(a) (b) (c)

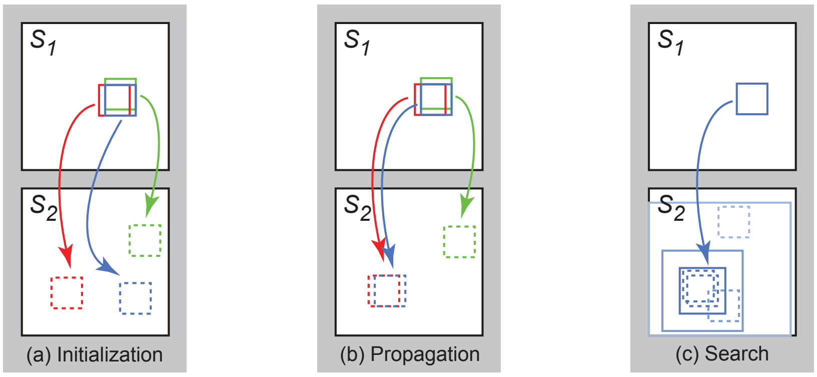
Figure 3.5: SIFT flow experiments: one frame from each of 731 videos was used as query image where the most similar was determined by the SIFT flow algorithm. (a): Query image. (b): Most similar image. (c) Warped onto (a) (figure taken from [38]).

- Instead of one, the number of detected nearest-neighbor patches is increased to k.
- The process of determining nearest-neighbors is extended by taking into consideration rotations and scales as well.
- For matching not only the distance metric SSD is used but also arbitrary descriptors and distances [4].

For a better understanding of the Generalized PatchMatch, PatchMatch is described in the following.

Let $\mathbf{R}$ be a patch in image S_1 and $f(\mathbf{R})$ the nearest-neighbor patch in image S_2. Then f denotes the Nearest-Neighbor Field (NNF) with dimensions of S_1 containing values called offsets. One possibility of filling the NNF is to choose randomly patches from S_2. Subsequently, the nearest-neighbor patch is optimized by executing propagation and random search.

In the propagation step for all patches it is assumed, that if the position of patches in S_1 is close to each other then neighbor patches in S_2 are likely close to each other as well. Considering a patch $\mathbf{R}$ and its left shifted patch $\mathbf{R}_{shift}$. Then in the propagation step it is attempted to improve the nearest-neighbor of $\mathbf{R}$ by substituting the patch one pixel to the right of $f(\mathbf{R}_{shift})$. If the mentioned patch offers a lower SSD to $\mathbf{R}$ it is registered as new nearest neighbor. Moreover, propagations are executed in iterations where even iterations use patches below and to the right and odd iterations neighbor patches above and to the left of $\mathbf{R}$. Executing propagation without random search would end up in a local minimum. Thus, in order to further reduce the distance to the nearest-neighbor patch a region around $f(\mathbf{R})$ is chosen where the minimal distance should be extracted in several iterations. This is called random search. Due to investigations of Barnes et al. the number of iterations are limited to a constant size [3]. Figure 3.6 illustrates the explained



(a) Initialization (b) Propagation (c) Search

Figure 3.6: Three steps of PatchMatch: (a) Initializing the NNF by choosing randomly nearest neighbor patches. (b) If the patch positions of image S_1 are close to each other the nearest neighbor patches should be close to each other as well. Thus, a possible new nearest neighbor for the blue patch is the red nearest neighbor, shifted one pixel to the right, and the green nearest neighbor shifted one pixel downwards. (c) Random search by searching a nearest neighbor in a fixed window size (figure taken from [3]).

steps of initialization, propagation and random search.

In Generalized PatchMatch the number of nearest-neighbors is extended to k. Therefore, the NNF, where heretofore for each pixel position one nearest-neighbor was stored, becomes a k-NNF. Now propagating patch $\mathbf{R}$ means testing k neighbors of $\mathbf{R}_{shift}$ which is shifted left, right, up and down. If one of the candidates is closer than the most distant patch associated to $f(\mathbf{R})$ the closer candidate is admitted to the field of k nearest neighbors.

In order to determine nearest-neighbors even though the patches are rotated or scaled around its center the search space is extended by these two dimensions. Hence the NNF is initialized by randomly choosing scale, orientation and positions in the given ranges. In the propagation step the transformation $\mathbf{T}(\mathbf{f}(\mathbf{R}))$, needs to be taken in consideration. $\mathbf{T}$ is determined over position, scale and orientations of the patches.

For the third improvement of using different distance metrics and descriptors the already mentioned hypothesis „*if the position of patches in S_1 is close to each other then neighbor patches in S_2 are likely close to each other as well*" is true as well [4]. Instead of using SSD as distance metric approaches are given in [4] where for symmetry detection a modified SSD metric per patch and in a label transferring application SIFT descriptors per pixel are used.

3.1.2 Optimization

One characteristic of optimization algorithms is the goal of minimizing errors and finding best solutions to a given problem. At the global minimum of so called energy functions the best solution is delivered and the returned error is at its lowest. An example of an energy function is the euclidean distance of two image patches' intensity values. To find the patch pair with the lowest euclidean distance of two images a naive method would be to compare one patch of image S_1 with each possible patch of image S_2. In case of additionally determining how well boundaries of adjacent patches fit together, it would end up in performance problems when iterating through each possibility of patch combination. For this optimization problem statistical approaches are introduced: a MRF delivers the best combination of image patches when more than one patch are available for a given part of an image which means that a joint probability distribution is found. Furthermore, the solution of a Poisson equation helps to find the best way of interpolating image regions. In this section the theory concerning the way a MRF and a Poisson equation can be solved is described in detail.

Markov Random Field

A discrete MRF belongs to the statistical family of graphical models. A graph is used to specify a family of probability distributions and gives the possibility to retrieve a joint probability distribution over all possible solutions [46]. An image can be partitioned in single segments (e.g. objects, regions or pixels) which are assigned to a label (e.g. color, depth or object class). Subsequently, a solution is found for the globally best combination of possible labels per segment and their conditional dependencies to each other (e.g. common boundaries).

A MRF is an undirected graphical model where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes the graph consisting of nodes $v \in \mathcal{V}$, representing image segments, and the connecting edges $\mathcal{E}$ in-between [46]. An undirected graph means that $\mathcal{E}(v_i, v_j) = \mathcal{E}(v_j, v_i)$. Two nodes $v_i, v_j \in \mathcal{V}$ are neighbors if they share a common edge $\mathcal{E}$ which in addition means that the neighborhood $\mathcal{N}$ of v_i covers all nodes $v_j \in \mathcal{V}$ that share an edge $\mathcal{E}(v_i, v_j)$ where $i \neq j$. The set of cliques originating from $\mathcal{G}$ is denoted as C. A clique $c \in \mathcal{C}$ is a subset of nodes from $\mathcal{G}$ forming a complete graph which means that there exists a connection between each possible pair of nodes. A maximal clique denotes the maximum number of nodes originating from the subset. An image with K segments holds K random variables **X** (one random variable per segment/node). All random variables together form a random field which contains the best combination (a.k.a. configuration and denoted by $\boldsymbol{\chi} = \{\chi_0, \chi_1, ..., \chi_K\}$) of predefined labels assigned to image segments. The joint probability is written as $\rho(\boldsymbol{\chi})$ indicating the probability of a possible configuration [46]. A graph is a MRF if:

- The probability that a random variable X_i takes a given label is larger than zero.
- Local Markovianity: dependencies exist only in the local neighborhood of a random variable. A local neighborhood in a graph is given only by directly connected nodes [46].

Equation 3.4 expresses these statements:

$$\begin{aligned} \rho(\chi_i) &> 0 \\ \rho(\chi_i | \boldsymbol{\chi}_{\mathcal{V}\backslash\{i\}}) &= \rho(\chi_i | \boldsymbol{\chi}_{\mathcal{N}(i)}). \end{aligned} \tag{3.4}$$

In the subset $\mathcal{V}\backslash\{i\}$ all nodes except i are included. In case of a local Markovianity the term $\rho(\boldsymbol{\chi})$ can be expressed as a Gibbs distribution to obtain the global joint distribution (and consequently the best configuration):

$$\rho(\boldsymbol{\chi}) = \frac{1}{Z} \exp \left\{ -\frac{1}{T} \sum_{C \in \mathcal{C}} V_C(\chi_C) \right\}. \tag{3.5}$$

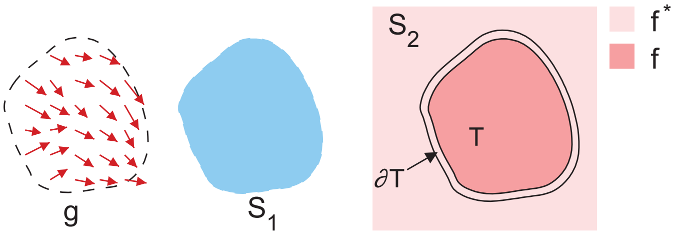


Figure 3.7: Sketch of pasting snippet S_1 on S_2. A seamless boundary of T and S_2 can be computed by solving the Poisson equation (adapted from [48]).

Z is a normalizing constant ensuring the sum to one and T a constant called temperature. A found clique in the graph is denoted by $\mathcal{C}$. V_C stands for a local function (a.k.a. clique potential) assigned to each maximal clique of the graph. In other words the global joint distribution is obtained by factorizing several local functions [46]. The MRF can also be seen as a minimization problem of the energy function $E(\boldsymbol{\chi})$:

$$\begin{aligned} E(\boldsymbol{\chi}) &= \sum_{C \in \mathcal{C}} V_C(\chi_C) \\ &= \sum_{\{i\} \in \mathcal{C}_1} V_1(\chi_i) + \sum_{\{i,j\} \in \mathcal{C}_2} V_2(\chi_i, \chi_j) + \sum_{\{i,j,h\} \in \mathcal{C}_3} V_3(\chi_i, \chi_j, \chi_h) + ... \end{aligned} \tag{3.6}$$

Equation 3.6 is the energy function E($\boldsymbol{\chi}$) for the clique potentials of pairs of neighboring pixels. For each clique an energy function exists which has to be minimized. In contrast, the global joint distribution has to be maximized to find the best configuration [46].

Poisson Equation

In [48] Pérez et al. present an approach for seamlessly stitching together different image contents. For image editing tasks - ranging from slight distortions of image regions over seamlessly stitching single images to panoramic views up to replacements of image regions - the Poisson equation can be used. The goal is to produce a smooth transition from one image content to the other by interpolating pixel intensities or colors originating from both the original and the new content.

Considering Figure 3.7, let S_2 be an image to paste on, T a closed subset of S_2 and target region and its boundary ∂T, f an unknown scalar function over the interior of T, f^* a known scalar function over the difference of S_2 and the interior of T, S_1 the snippet to paste and g a guidance field (a.k.a. vector field) of S_1. The goal is to interpolate S_1 and place it on S_2 without observing any boundaries or disturbing seams between those regions. The basic idea behind solving the Poisson equation is to define the function f over T which gets constrained by forcing the function to take the values of f^* at the common boundary ∂T.

To avoid blurred interpolants Pérez et al. propose a guidance field g which is part of the minimization problem:

$$min_f \iint_T |\Delta f - g|^2 \text{ with } f|_{\partial T} = f^*|_{\partial T}, \tag{3.7}$$

where Δ is the Laplacian operator. The guidance field g is the gradient field of S_1. The process of minimization has to satisfy the Poisson equation

$$\Delta f = \text{div} g \text{ over } T, \text{ with } f|_{\partial T} = f^*|_{\partial T} \tag{3.8}$$

which means that div$g$ is valid over the region defined by T [48]. The term div denotes the divergence of g. Additionally, a Dirichlet boundary condition is valid giving the information of what the values of f should be at given positions. In this case the given positions are $(f|_{\partial T} = f^*|_{\partial T})$ saying that on ∂T the values of f equals the values of f^*. In other words, the main problem for solving the Poisson equation is to find a correction function $\hat{f}$ on T such that $f = \hat{f} + S_1$ [48].

3.2 Morphing Framework

This section describes the technical workflow of morphing multiple images. For a better understanding, this workflow is visualized in Figure 3.8. Initially, the source images have to be registered to each other. Therefore, the SIFT flow algorithm is used to find correspondences in images[1]. For each pair of images a displacement field is computed. A so called best reference image is chosen by finding the minimum of displacements summed over all related displacement fields where 'related' means images where correspondences exist. The images are aligned by a warping process, by shifting pixels of images according to its displacement field which points to the best reference image. Visually common regions are valuable and have to be preserved. These regions are represented as rectangular patches and included in a synthetically produced central image. According to this template the morphing process reproduces a visually similar output by using two optimization frameworks. Firstly, a MRF finds the best combination of similar patches compared to the central image and the smallest visual difference between patch boundaries of the output image. Secondly, the solution of a Poisson equation smooths the morphed image such that any boundaries or disturbing seams become invisible.

3.2.1 Image Registration

The first step of the morphing framework consists of examining the relationship of the source image's content where $S = \{S_1, S_2, ..., S_K\}$ and $K > 2$. Based on these relationships, subsequently the images are warped which means that the image's content gets aligned to each other. The task of finding correspondences is realized by the SIFT flow algorithm which is described in detail in Section 3.1.1. The extracted SIFT features are invariant to scale and partially invariant to changes of illumination and 3D viewpoints [39]. Concentrating on the recognition of coin images, the evaluation of Kampel and Zaharieva shows outstanding performances of SIFT features [31]. SIFT flow computes a dense correspondence field where correspondences are found by optimizing the energy function comprising the data term (Equation 3.1), small displacement term (Equation 3.2) and the smoothness term (Equation 3.3).

---

[1] The implementation of SIFT flow is supported by using the SIFT flow library, available at `http://people.csail.mit.edu/celiu/SIFTflow/` (accessed on 03.10.2013)

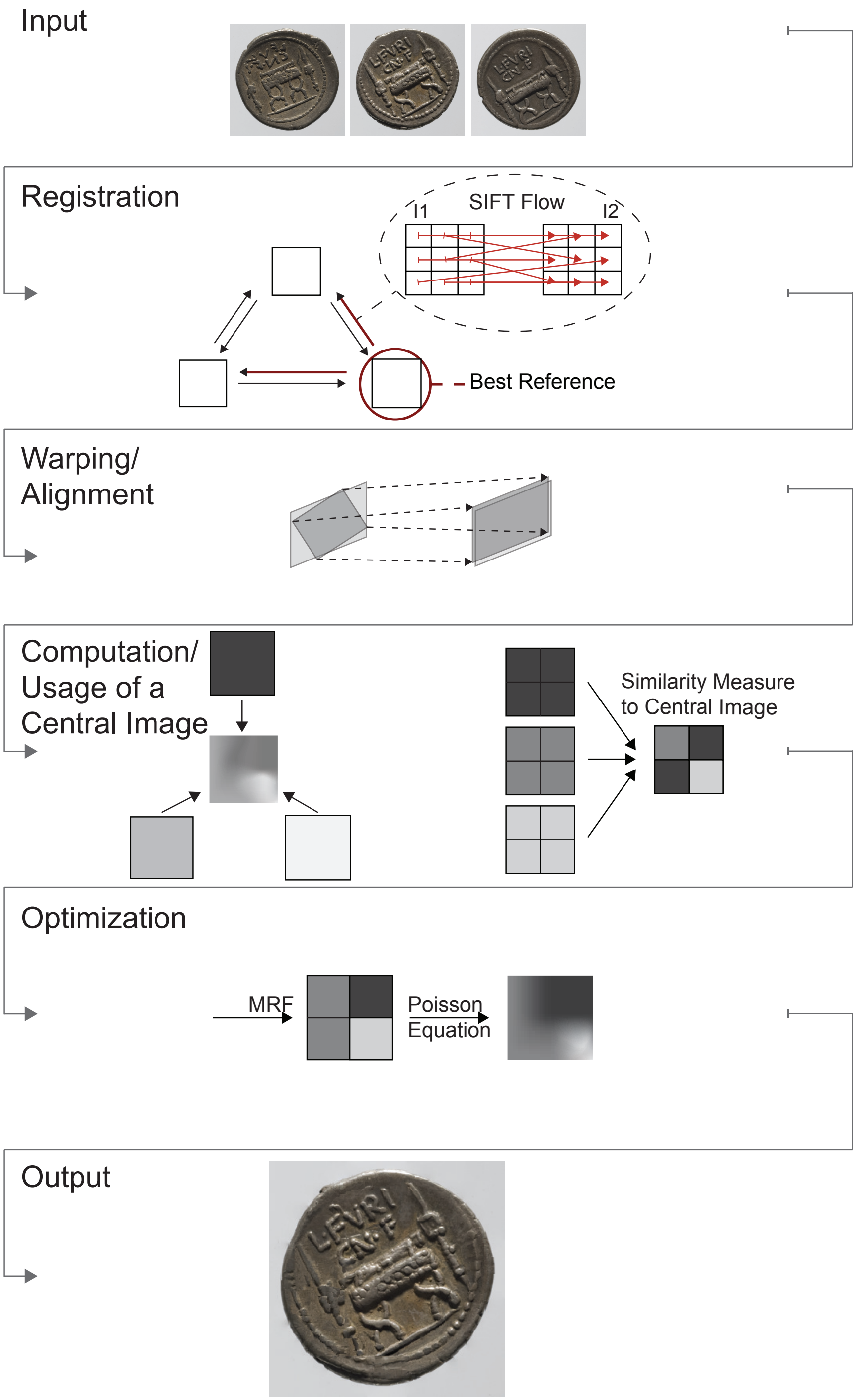
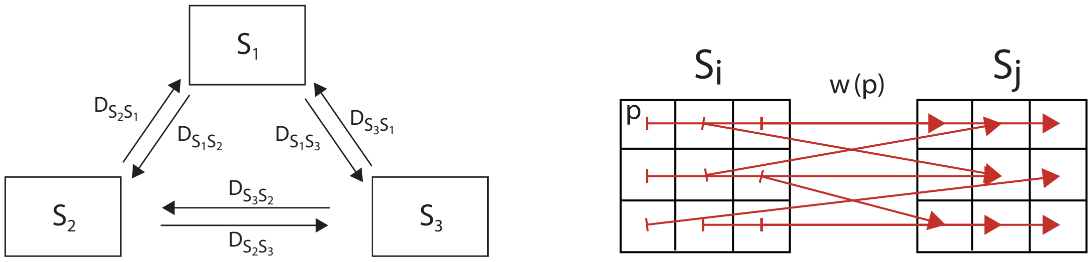


Figure 3.8: Workflow of the multi-image morphing framework from upper left to bottom right.



(a) (b)

Figure 3.9: (a) S_1, S_2 are the given source images. The arrows in-between represent displacement fields D containing flow vectors for each pixel. (b) An example of a displacement field. The red arrows indicate the flow vectors $\mathbf{w}(p)$ which are calculated for each pixel position p and point to one corresponding position in S_j.

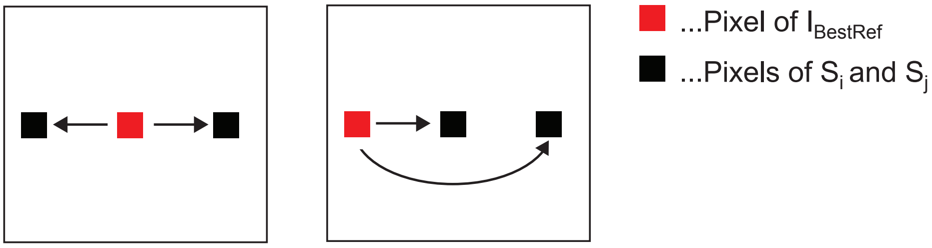
For each pair of source images a pixel displacement field D is computed. In the following p defines a pixel in S_i. Let $\mathbf{w}(p)$ be the flow vector determining the corresponding pixel of p then D holds for each pixel a flow vector. Consequently, $K \cdot (K-1)$ displacement fields are generated where each field originates from one reference image serving as basis for the calculation of the flow vectors. Figure 3.9a gives an example of the relationship among source images. Each source image can take the role of a reference image. Stored flow vectors for each pixel form the displacement field $D_{S_iS_j}$ from S_i to S_j. Figure 3.9b shows an example of a displacement field where S_i and S_j are the source images.

Since for further computations only one image serves as reference image a decision has to be made which image should be selected. Once all displacement fields are calculated, the reference image is chosen by finding the minimum of displacements over all related displacement fields. In Figure 3.9a e.g. a related displacement field of S_1 is $D_{S_1S_2}$ and $D_{S_1S_3}$. If the displacements originating from S_1 are smaller than those of S_2 and S_3 then S_1 is selected as I_{BestRef}. For a better understanding in Figure 3.10 an example of the selection is given. For simplicity, it is assumed that all images are put on top of each other. The red pixel is chosen to be from I_{BestRef} and the other pixels are from S_i and S_j. In 3.10a it can be seen that a correct selection of I_{BestRef} is done, since the total distance to all shifted pixels is at a minimum. In contrast, in 3.10b an example is shown where the distance is increased by choosing a wrong I_{BestRef}. Since I_{BestRef} is defined by selecting the minimum of displacements over all related displacement fields, the content of the reference image is the closest to an „average content“.

3.2.2 Warping

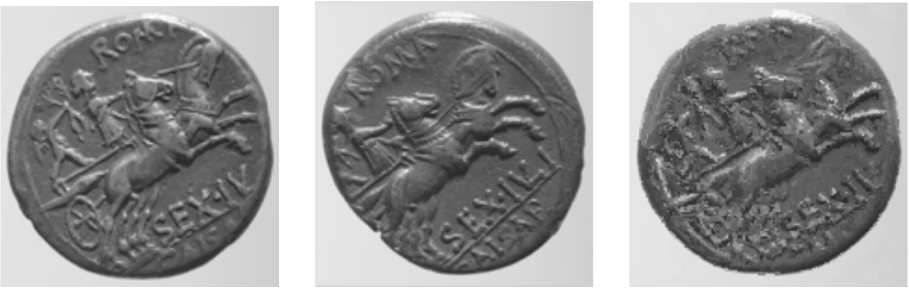
In [38] Liu et al. use the pixel displacement field as a warping function to investigate the quality of the function. Matched images of the database are warped onto the query image. Examples are given in Figure 3.5c which are warped onto the images in Figure 3.5a.

For this work the displacement field is also seen as a warping function. Let q be a pixel of S_j at position (x, y) and $\mathbf{w}(q) = \big(\mathbf{u}(q), \mathbf{v}(q)\big)$ the flow vector split in displacements in x- and


🟥 ...Pixel of I_{BestRef}
■ ...Pixels of S_i and S_j

(a) (b)

Figure 3.10: (a) A correct selection of I_{BestRef} with a minimum distance to S_i and S_j. (b) An example of a wrong selection of I_{BestRef}, since the total distance is higher than in (a).



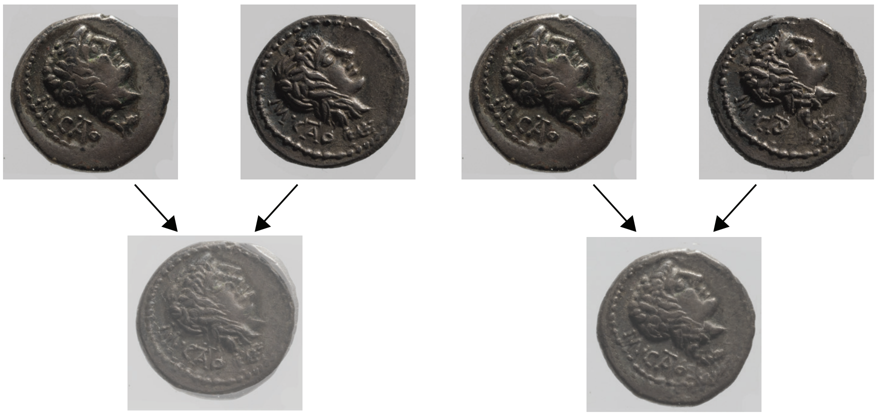
(a) (b) (c)

Figure 3.11: (a) Query image. (b) A coin image from the same class is used. (c) Warping image of Figure (b) onto (a) (figure taken from [72]).

y-direction. If a displacement field exists of I_{BestRef} and S_j then pixel q is shifted by $\mathbf{u}(q)$ in x-direction and $\mathbf{v}(q)$ in y-direction to finally obtain the warped images $I'_1, I'_2, ..., I'_K$. In Figure 3.11 the displacement field of two coins belonging to the same class is computed to subsequently warp the image in Figure 3.11b onto the query image of Figure 3.11a. The warped image is shown in 3.11c. For a better understanding, Figure 3.12 compares two different cases where on the one hand the given images are not aligned (Figure 3.12a) and on the other hand they are aligned (which means that one image is warped onto the other; Figure 3.12b). This example indicates why the alignment of image contents is needed as a fundamental step for image morphing.

3.2.3 Computation and Usage of a Central Image

Considering the resulting morphed image, the objective is to summarize relevant visual data from all source images. For this work visual data of a source image is relevant if it has visual similarities to other source images. Visual regions which match to several other images are considered as valuable and have to be preserved while non-matchable regions have to be eliminated. These constraints are realized by introducing a central image with the purpose of having a template which comprises common regions of images. The more images exhibit a common region the clearer the illustration of the region appears. Let I_C be the central image, then the



(a) (b)

Figure 3.12: (a) Two images which are not aligned and put on top of each other. (b) The same two images aligned.

computation can be realized by determining the mean of the pixel values at each pixel position of all warped images. This can be written as:

$$I_C(p) = \frac{1}{K}\sum_{i=1}^{K} I'_i(p). \tag{3.9}$$

Another option is to calculate the median of a sorted set of pixel values, having the advantage of being less sensitive to outliers:

$$I_C(p) = \begin{cases} I'_{\frac{(K+1)}{2}}(p) & K \text{ odd} \\ \frac{1}{2}\{I'_{\frac{K}{2}}(p) + I'_{\frac{K}{2}+1}(p)\} & K \text{ even} \end{cases} \tag{3.10}$$

Figure 3.13 illustrates these two possibilities of computing a central image. In 3.13b the image is generated by using Equation 3.9 and outliers are still visible. On the contrary, in Figure 3.13c Equation 3.10 is employed in order to compute I_C with four source images. This method clearly outsources artifacts illustrated within the red ellipses.

Since the contours and structures of I_C can be blurred or hardly visible, the target image T is visually approximated in order to obtain a sharp output. Therefore, I'_i and I_C are divided into patches where each one represents a rectangular region of pixels and can be controlled separately. Let R be a patch of image I'_i and Q denotes a patch of I'_C then patch R is compared to Q. This is done for all patches. It should be emphasized that a comparison of the patches is only done if both share a the patch position in the image. This process serves to determine the similarity of each patch originating from I'_i and I_C, respectively. The result of the similarity measure is distance dP for each patch.

For the measurement of dP two implementations exist:

- Firstly, SIFT flow offers an energy function (consisting of the terms, described in Equations 3.1, 3.2 and 3.3) which can be exploited to estimate the visual similarity of the

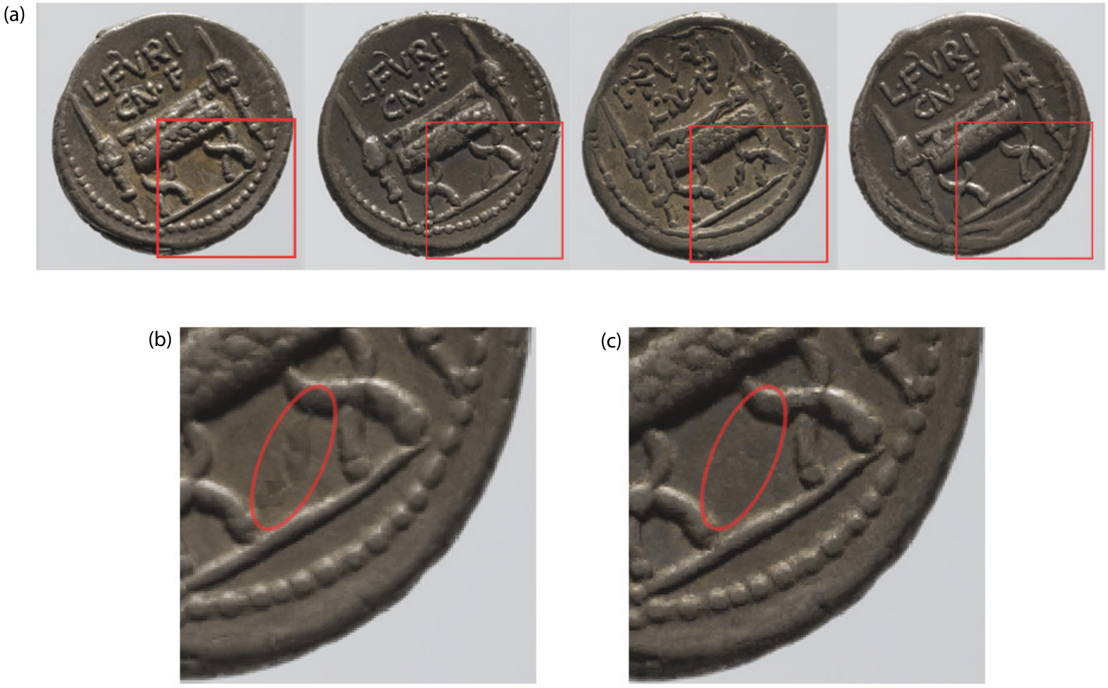


Figure 3.13: (a) Source images I'_1, I'_2, I'_3, I'_4. (b) Mean of all input images. (c) Median of all input images.

patches. Zambanini et al. present this procedure in [72] to obtain the visual similarity of two given coin images.

- Secondly, for each patch a SIFT descriptor can be computed where its width equals the with of a patch. Let $D1$ denote a descriptor of patch R and $D2$ the descriptor of patch Q then dP is received by calculating the l_2-norm.

Since for this work the usage of the energy computed by SIFT flow causes a lack of performance the descriptor method is used. Finally, for each patch position of target image T a patch is selected where the distance to Q is at a minimum.

3.2.4 Optimization

As described in Section 3.1.2, an important characteristic of optimization methods is the objective of improving already existing solutions. The visual appearance of the target image T can be optimized by using probability functions for the selection process of an image patch and finding the best combination of patches by exploiting a MRF. The combination of patches is computed by selecting the patch holding the shortest distance to the patch of the central image. In other words the selection of a patch is restricted to one local similarity criteria. In order to visually improve the result such that the number of abrupt boundaries between patches is reduced neighbors of the patch are included in the selection process. This means that not only the patch with the highest similarity is chosen but also a transition as smooth as possible between patches is a decisive factor. In order to find the solution with the maximum likelihood for the combination of patches where the similarity to the central image is as high as possible and the transition to its neighbor patches is as smooth as possible a MRF is exploited. Additionally, still disruptive boundaries between selected patches are smoothed by solving the Poisson equation. Therefore, intensity differences of the selected patches are corrected by interpolating pixel intensities according to values of its adjacent patches to reach a homogenous appearance of the image.

Find Best Patch Combination Using Markov Random Field

Given are K images where each one consists of L x M patches. The target image T can be improved by additionally minimizing the visual difference of adjacent patch boundaries. Thus, the selection of each patch is dependent on two conditions, namely the similarity to the central images' patch and to its connected neighbors. Firstly, values of the patch-based similarity measurement have to be transformed to probabilities. For each patch R, a distance dP to patch Q is calculated where both patches share a patch position. Similar to [75], dP is transformed to a pseudo probability $\rho(dP)$ where the shortest distance is mapped to the highest probability and vice versa. Consequently, $\sum_{i=1}^{K} \rho(dP_i) = 1$. To regulate the ratio of the highest and the lowest probability, the parameter θ is introduced: the probability of the lowest distance is θ times higher than the probability with the highest distance. Consequently, it follows

$$\tilde{dP} = \frac{dP - \min_{(l,m)}(dP)}{\max_{(l,m)}(dP) - \min_{(l,m)}(dP)}(\theta - 1) + 1. \tag{3.11}$$

Finally, the similarity probability is given by

$$\rho(dP_i) = \frac{1}{\tilde{dP} \cdot \sum 1/\tilde{dP}} \tag{3.12}$$

Once the probabilities of the patch similarity measurement are computed, the optimization continues by introducing the Adjacent Patch-Boundary Disparity (APBD). APBD denotes the quality of a transition between adjacent patches. Considering Figure 3.14a, an adjacent patch can either be in the 4-connected neighborhood of the same level (same image) or in the 4-connected neighborhood of different levels, as can be seen in Figure 3.14b.

For simplicity two adjacent patches are denoted as R and Q. For the case of considering two vertically adjacent patches with height H and width W the calculation of distance dE is started by choosing two rows per patch at the border of R and Q. Measuring the quality of transition between patches is done by setting up a Sobel operator [19]. The top row $R(H)$ and the row nearest to the bottom $Q(1)$ are selected. In order to compare the gradients of both rows, as a pre-processing step the convolution kernel $\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$ is applied to each of them. Finally,

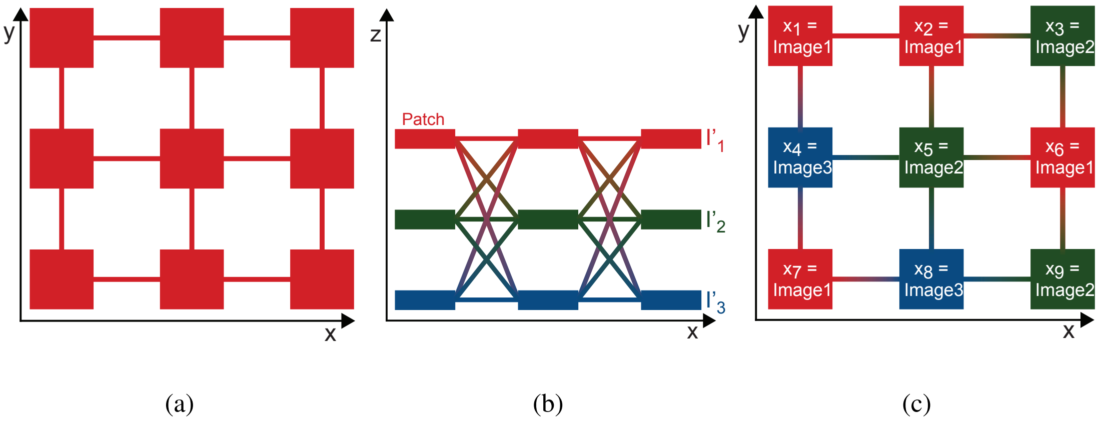


Figure 3.14: All patches originating from a common image are colored unitary. (a) Concept, seen along the z-axis where I'_1 is divided into 3×3 patches. (b) Concept, seen along the y-axis where I'_1, I'_2 and I'_3 arranged on top of each other. (c) A possible MAP configuration found by a MRF where each member of the configuration $\mathbf{x} = \{x_1, x_2, ..., x_{L \times M}\}$ holds one label per patch position.

the searched distance dE is defined by

$$\begin{aligned} dE = &|| \begin{bmatrix} -1 & -2 & -1 \end{bmatrix} * R(H) + \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * Q(1) ||_1 + \\ &|| \begin{bmatrix} -1 & -2 & -1 \end{bmatrix} * R(H-1) + \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * Q(2) ||_1 \end{aligned} \tag{3.13}$$

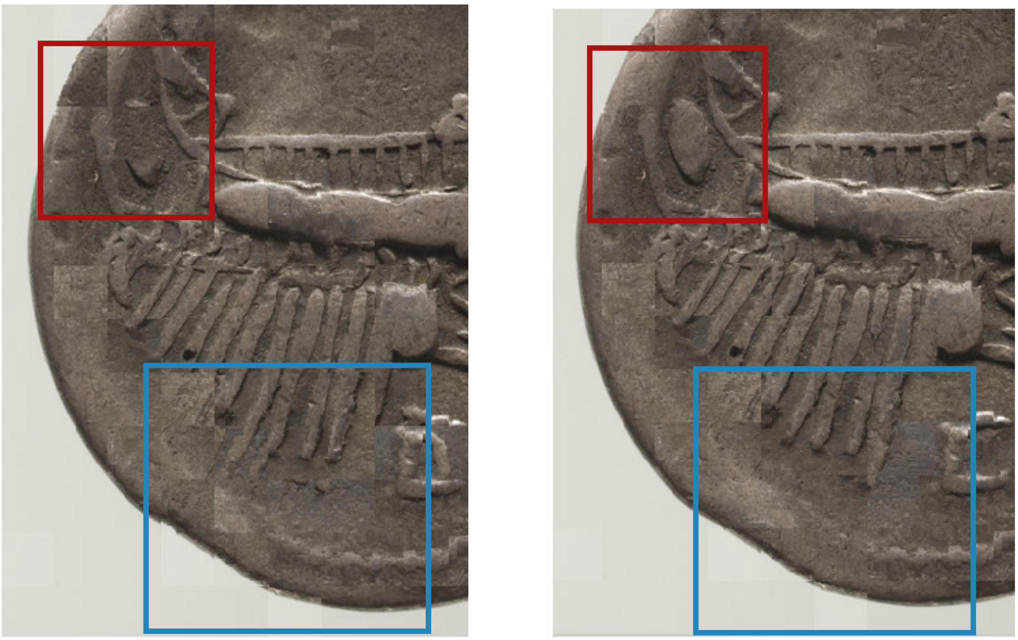
where $*$ stands for convolution and $|| \cdot ||_1$ denotes the l_1-norm. The objective of this equation is to highlight well fitting structures and in the same it is invariant to offsets or in other words color differences of the given patch boundaries are neglected.

To obtain the probability of dE Equations 3.11 and 3.12 can be used. A visual concept of the arrangement of the patches is illustrated in Figure 3.14. Each layer represents one image consisting of L x M patches. Each patch holds one probability ρ. In the interests of simplification, the whole concept is represented from two different points of view. All patches originating from a common image are colored unitary. Edges stand for direct dependencies between patches holding the probability resulting from the APBD approach. For each patch position of the target image one patch of either $I'_1, I'_2, ..., I'_K$ is selected. To find the best combination of patches for the target image where the global probability is at a maximum, a MRF is used[2]. The best combination of patches is called Maximum a-posteriori (MAP) configuration.

For a MRF, $\mathcal{V}$ represents $L \times M$ patch positions. The variable $\mathcal{E}$ holds all edges of the graph and determines the neighborhood of the vertices. The set of labels is given by $\mathcal{L} = \{I'_1, I'_2, ..., I'_K\}$. To each patch position one random variable $\mathbf{X}_i \in \mathcal{L}$ is assigned. The best configuration χ consists of $L \times M$ labels. The energy term of the MRF is defined by a combination of unary and

---

[2] The implementation of the MRF is supported by using the Undirected Graphical Models (UGM) library, available online: `http://www.di.ens.fr/~mschmidt/Software/UGM.html` (accessed on 16.12.2013)



(a) Simply patches with the shortest distance to central image patches are selected.

(b) A MRF optimization is used for patch selection.

Figure 3.15: Example of a MRF optimization. The red and blue rectangles are highlighting regions where the transition between image patches is improved.

pairwise potentials

$$E(\boldsymbol{\chi}) = \sum_{i \in \mathcal{V}} \phi_i(\chi_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}(i)} \phi_{i,j}(\chi_i, \chi_j), \tag{3.14}$$

The pairwise potential includes all neighbors (4-connected neighborhood) of node i and is defined by

$$\phi_{i,j}(\chi_i, \chi_j) = \rho(dE). \tag{3.15}$$

The unary potential is defined by

$$\phi_i(\chi_i) = \rho(dP) \tag{3.16}$$

and the MAP configuration $\hat{\boldsymbol{\chi}}$ is computed by

$$\hat{\boldsymbol{\chi}} = \underset{\boldsymbol{\chi}}{\text{argmin}}\ E(\boldsymbol{\chi}). \tag{3.17}$$

Figure 3.14c shows a possible target image where for each position of the image one patch is selected and is as a member of the final configuration set $\boldsymbol{\chi}$ with the highest probability. Figure 3.15 illustrates the improvement of the image patch selection. In Figure 3.15a patches holding the shortest distance to the central image's descriptor are selected. In contrast, in Figure 3.15b the patches are selected by using Equation 3.14 and applying the MRF optimization. In order to demonstrate a possible result after applying the MRF, in Figure 3.16 the source images I'_1,

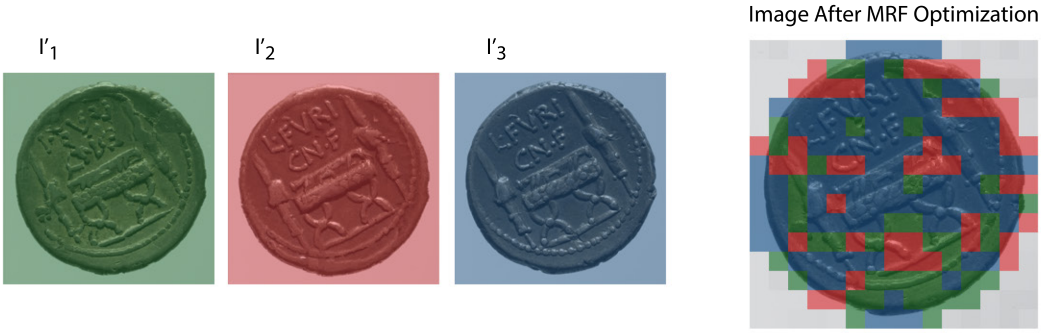


Figure 3.16: The patch configuration after applying a MRF is colored according to its originating images.

I'_2 and I'_3 are colored green, red and blue, respectively. The resulting image located at the right comprises patches from each input image. Each patch is colored according to its originating image.

Smoothing Disruptive Patch-Boundaries by Solving the Poisson Equation

In Section 3.1.2, the theoretical background of solving the Poisson equation is explained. This subsection describes a discrete Poisson solver and additionally explains the use of the solver in the implementation of this thesis. For this purpose S_2 and T are a set (or a subset) of pixels. Additionally, N_p is introduced as the 4-connected neighborhood of pixel p where $|N_p|$ can be smaller than 4 at borders of the pixel grid. If q is an element of the neighborhood of p then the minimization problem can be written as

$$\min_{f|_T} \sum_{(p,q)\cap T\neq\emptyset} (f_p - f_q - g_{pq}), \text{ with } f_p = f^*_p, \text{ for all } p \in \partial T, \tag{3.18}$$

where g_{pq} are the gradients of pq. The solution has to satisfy the equation

$$\text{for all } p \in T,\ |N_p| f_p - \sum_{q\in N_p\cap T} f_q = \sum_{q\in N_p\cap\partial T} {f_q}^* + \sum_{q\in N_p} g_{pq}. \tag{3.19}$$

where the contribution of the first term on the right hand side is constrained to pixels at the boundary of T [48].

For the implementation an adaption of the seamless tiling approach in [48] is made. The intention of this task is to stitch together the selected patches such that any boundaries or disturbing seams in-between become invisible. Initially, the problem can be considered as illustrated on the left hand side of Figure 3.17 where two adjacent patches R and Q share a common boundary. In order to get a smooth transition from R to Q, first Q is considered. The region of Q is compared to „the snippet to paste“ and becomes S_1. Region S_2, where the patch should get pasted onto, is illustrated at the right hand side of Figure 3.17. It can be seen that T has the size of S_2 and ∂T is divided into ∂T_C and ∂T_B, whereby conditions for ∂T_C are defined by taking the mean

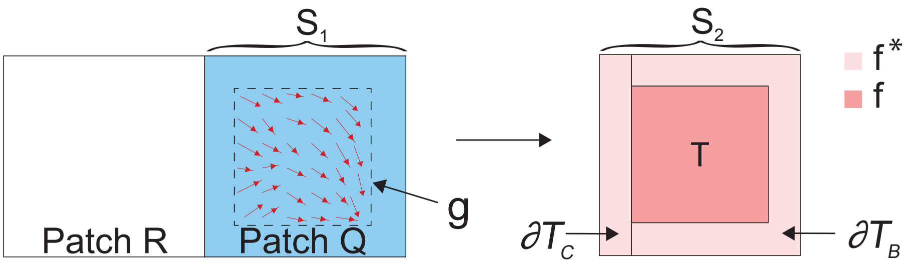


Figure 3.17: Solving the Poisson equation of adjacent patches R and Q. While T is the region to interpolate ∂T_C and ∂T_B denotes the boundary of the patch which is well known. ∂T_C is calculated by the shared boundary pixel values of R and Q whereas ∂T_B comprises only boundary pixel values of Q.

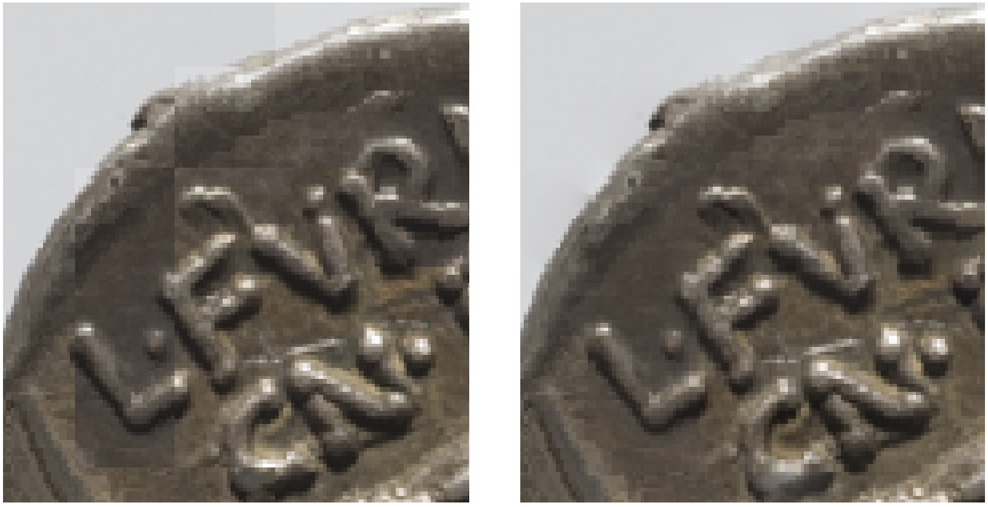


Figure 3.18: The Poisson optimization is applied to the image on the left hand side. On the right hand side the result of removed boundaries between the image patches is illustrated.

pixel values of the rows at the shared boundary of R and Q. For ∂T_B the values at the boundary of Q are used. The known function f^* is given at ∂T and f is defined over T. The vector field g is calculated by the convolution on the Laplacian operator and Q over the interior of T. The same process has to be done for patch R. The result of removing disturbing seams is shown in Figure 3.18 where on the right hand side the patch colors are interpolated by using the Poisson equation approach.

3.2.5 Demonstration on a Synthetic Data Set

In order to communicate the two basic ideas of the morphing framework, it is applied to synthetic images. The first example illustrates the intention of preserving visually common regions of the source images. Three squares serve as source images (illustrated in Figure 3.19). Their visual appearance is slightly different: while the contour of S_3 is preserved completely, the images

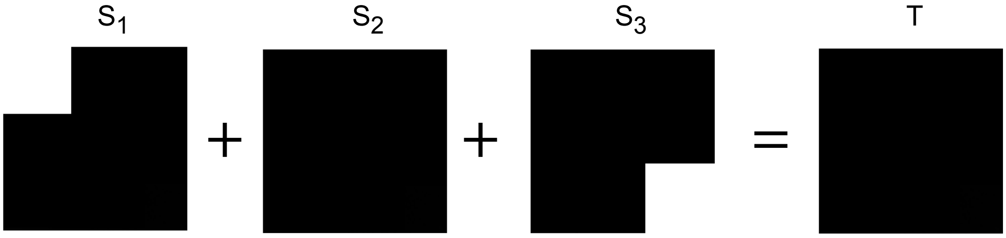


Figure 3.19: Applying the morphing framework on three slightly different looking squares: the clipped corners are eliminated due to the fact that they appear only once. In contrast, all black regions appear multiple times.

S_2 and S_1 show clipped corners. Morphing all three squares results in a completely preserved square T. This example illustrates the elimination of invaluable data (in this case the clipped corners of S_2 and S_1) by only preserving regions which appear multiple times in different images (all black regions of S_1, S_2 and S_3). Conducted on coin images, this technique helps to eliminate unique appearances of coins like damages which are not valuable for representing a class of a coin. The second example is based on hand-written signatures (one per source image). Figure 3.20 depicts six signature images $S_1, S_2, ..., S_6$ which are warped onto I_{BestRef}. Image T at the bottom shows the result after going through the morphing pipeline. Due to natural conditions all signatures are slightly different in their appearance. In consequence of the image simplicity, after warping onto the reference images, the contour of the warped images are close to I_{BestRef}. Nevertheless, since I_{BestRef} is defined by selecting the minimum of displacements over all related displacement fields, the contours of the reference image is the closest to an 'average contour'.

3.3 Summary

In this chapter a multi-image morphing approach is presented where the SIFT flow algorithm acts as core element. Further image registration methods such as PatchMatch or optical flow methods are discussed in detail as well as the functionality of SIFT flow and its underlying SIFT algorithm. SIFT flow uses SIFT descriptors to describe extracted features which are compared and rated. The most similar features form a pixel correspondence. The computed pixel correspondences are exploited within a warping process where the resulting images show one uniform alignment. The objective of outsourcing unitary regions and retaining common regions is fulfilled by introducing a central image whose content can be generated by calculating the mean of multiple source images. Improvements of the resulting visually summarized image are achieved by exploiting a MRF and finally solving the Poisson equation for intensity corrections.

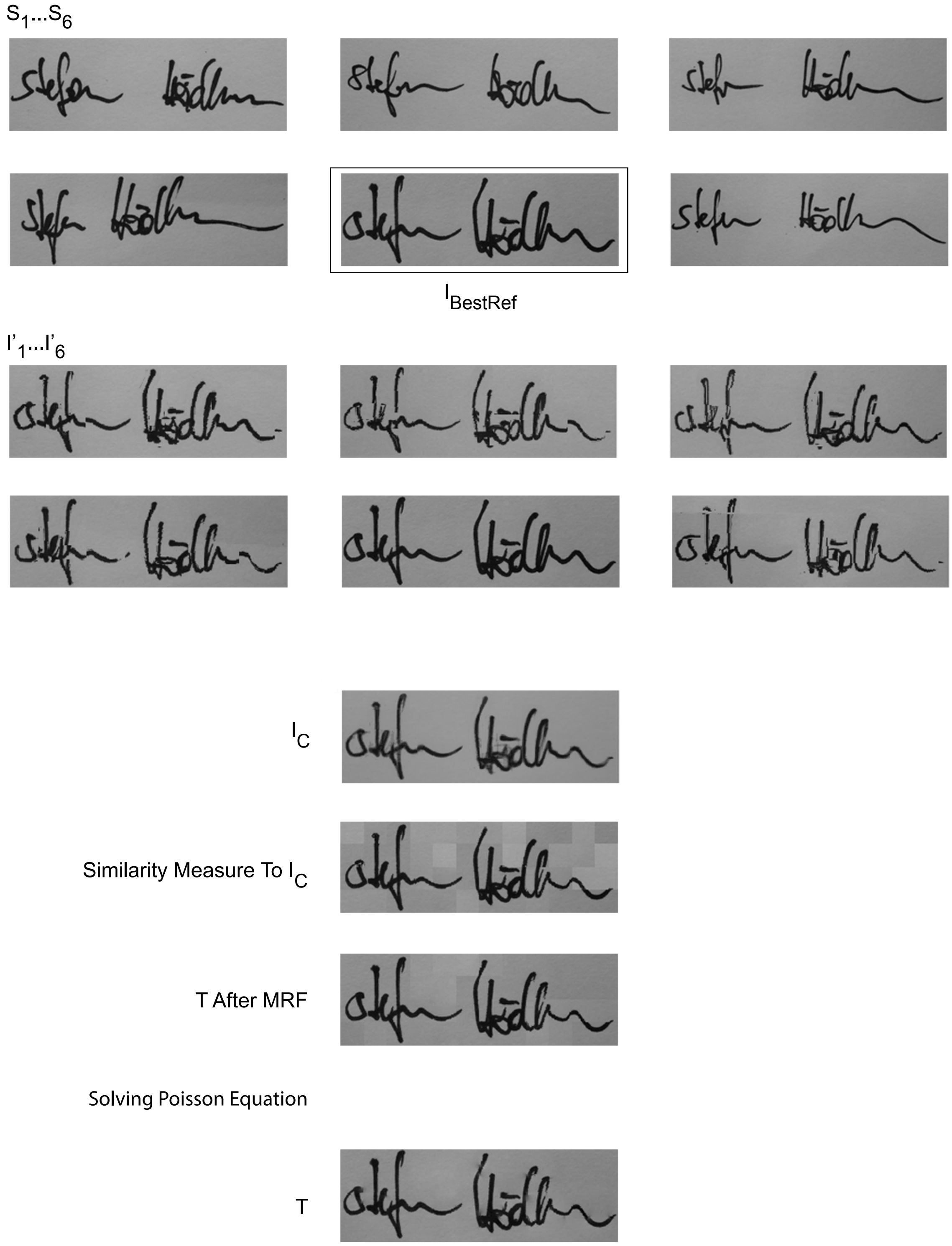


Figure 3.20: Six different signature styles used as source images for the morphing framework.

CHAPTER 4

Experiments

In this chapter the algorithm introduced is evaluated. This is done by dividing the evaluation in two main parts. Firstly, in the context of qualitative experiments the morphing results are examined according to visual appearance. The output is generated by empirically evaluating the algorithm's parameters. In order to show that an aesthetically morphed image can not only be achieved on ancient coin images the morphing framework is applied on two more data sets: a data set containing images of airplanes and a data set of gray scale images illustrating different appearances of human faces.

The second part of this chapter comprises the quantitative experiments and deals with the examination of the morphing results according to its possible applications in computer vision. This evaluation is executed by means of a classification approach of ancient coins. In the process of assigning classes to coin images the morphed image serves as training image. An optimized parameter configuration of a morphed image is found by using the ROC analysis. Finally, it is shown that using a morphed image instead of a single image as training image leads to a higher classification rate.

As an introduction, Section 4.1 presents all data sets used for the evaluations. Subsequently, qualitative experiments described in Section 4.2 show the strengths and drawbacks of the morphing algorithm conducted on all proposed data sets. Quantitative Experiments, which describe the usage of a morphed image in a classification task, and the parameter optimization using a ROC curve are presented in Section 4.3. A brief discussion with a comparison of qualitative and quantitative results is given in Section 4.4. Finally, a summary in Section 4.5 concludes the chapter.

4.1 Data Sets

This section presents the data sets used for the proposed algorithm of this Master's thesis. In order to demonstrate the adaptability of the morphing framework it is not only conducted on a coin data set but also on other data sets. Two more data sets are selected to present the adaptability: a data set of airplanes in side view and the Yale Face data set.

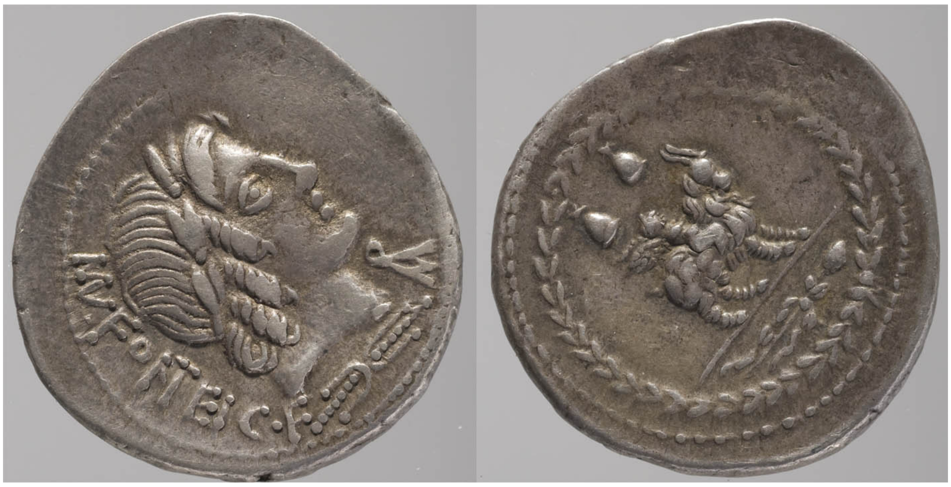


Figure 4.1: An example of obverse side and reverse side listed with reference number 153/1a in [14].

Coin Data Set The coin data set comprises coin images of the Roman Republican coinage. The coins are made available by the Coin Cabinet of the Museum of Fine Arts in Vienna, Austria [72]. All classes of the data set hold a reference number defined by Crawford's standard reference book [14]. In Figure 4.1 an example is given of the obverse and reverse side belonging to the coin with the reference number 153/1a listed in Crawford's catalogue. The illustrated coin was minted in Rome about 85 BC and weighs 3.9 grams. The image on the obverse side shows the letters *MN FONTEI C F*. The description in the catalogue reads: *„A laureate head of Vejovis (or Apollo) wearing hair in loose locks on the right. Below a thunderbolt and a Roma monogram is illustrated. On the right hand side the reverse side shows two infant winged Genius (or Cupid) seated on a goat. Caps of the Dioscuri are above. A thyrsus with fillet in the exergue and all is located within a laurel wreath"* [14]. With the help of a description of that kind coins can be classified manually. In Figure 4.2 the data set used is shown which comprises 50 classes of ancient coins. Class 37 and 38 depicts different classes but similar scenes. This can be also shown in class 29 and 30. The coins might be worn or damaged but the degree of preservation can change between individual specimens of a coin type and even locally on a single coin as can be seen in class 41 where the coin comprises a hole.

Airplane Data Set The second data set, composed by the California Institute of Technology, consists of 1074 color images illustrating airplanes which are all photographed sideways. As an example in Figure 4.3, 24 images with varying resolution are presented. The data set is available online[1].

Yale Face Data Set The third data set used for this Master's thesis comprises 164 gray scale images of human faces with a size of 300 x 300 pixels. 11 different persons are portrayed

---

[1] `http://www.vision.caltech.edu/archive.html` (accessed on 02.04.2014)

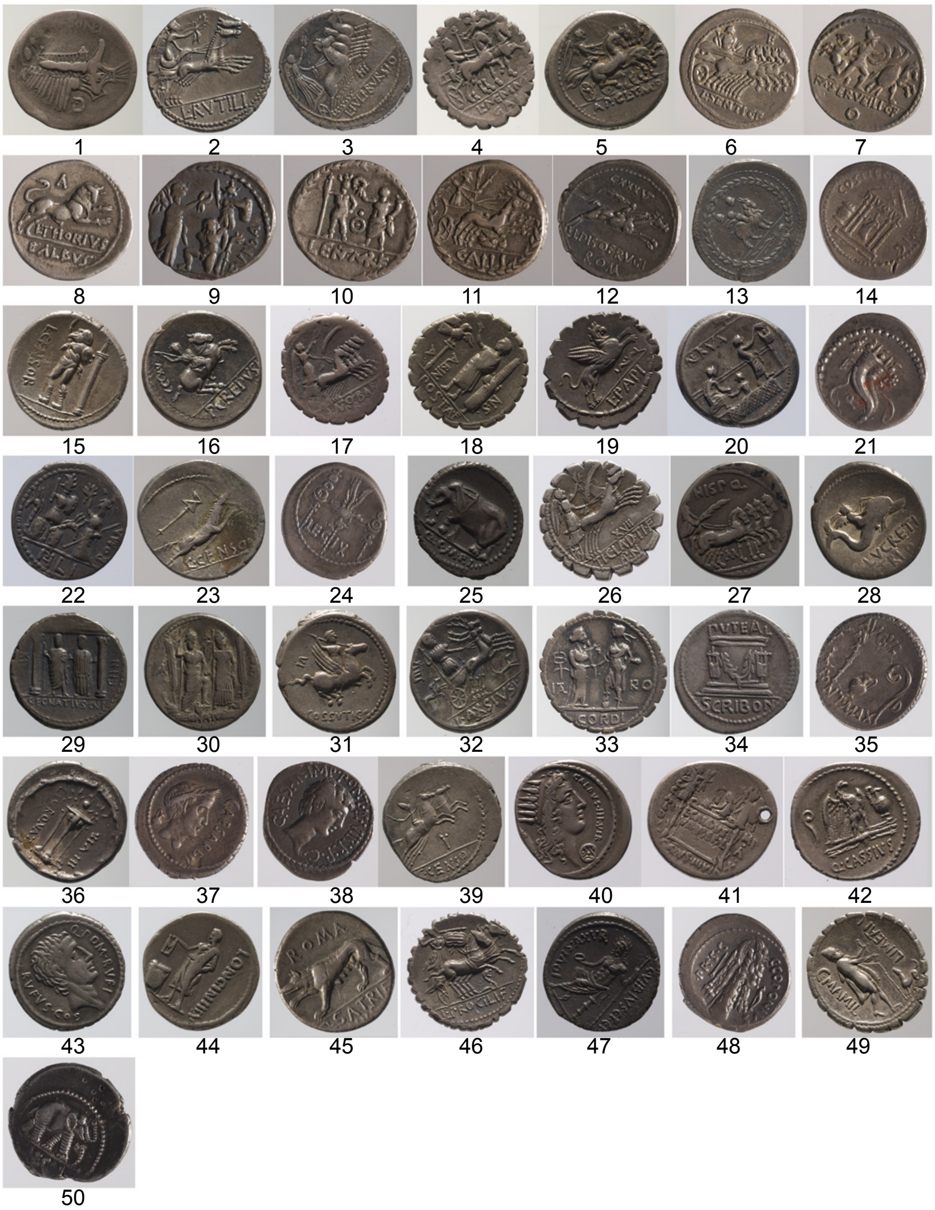


Figure 4.2: 50 reverse side classes of the ancient coin data set.

with different appearances which are: „left-light“, „center-light“, „right-light“, „with glasses“, „without glasses“, „happy“, „normal“, „sad“, „sleepy“, „surprised“ and „wink“. The background of all images is almost white. Figure 4.4 shows an excerpt of eight persons each one presenting „without glasses“, „happy“ and „with glasses“. This data set is described in [6] and is available online[2].

---

[2] `http://vision.ucsd.edu/content/yale-face-database` (accessed on 02.04.2014)

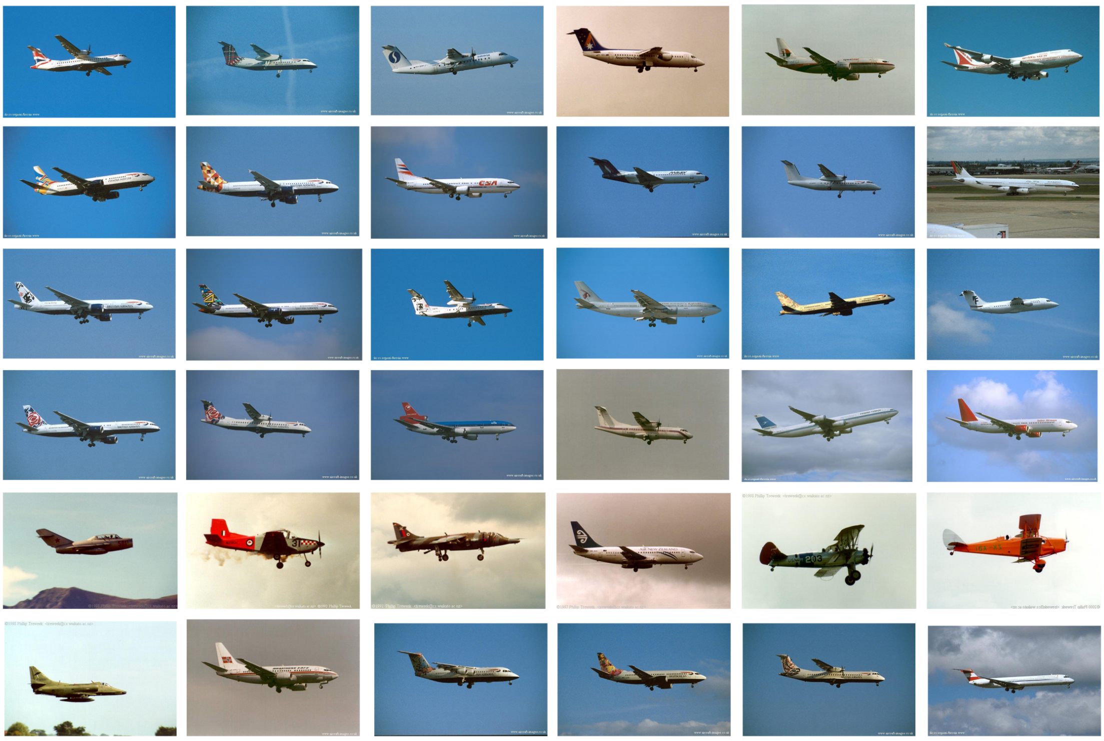


Figure 4.3: An excerpt of the airplane data set.

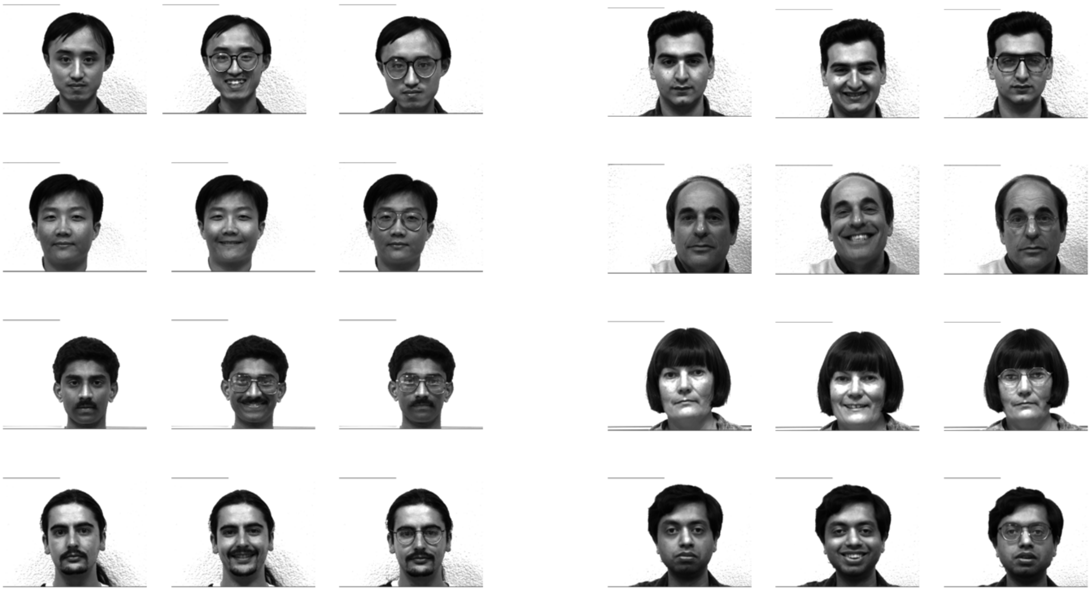


Figure 4.4: An excerpt of the Yale Face data set.

4.2 Qualitative Experiments

Qualitative experiments are conducted on all three data sets described in Section 4.1. The objective of this section is to generate an image T using three source images S_1, S_2, S_3 which are assumed to have similar orientations. Results of the qualitative experiments demonstrate the capabilities of the framework concerning solely the visual appearance of T. Resulting images should be visually complete and coherent which means that as much as possible common regions of S_i are summarized in one image and at the same time T should appear aesthetically without exhibiting any disturbing artifacts. The parameters used for the algorithm remain the same for all data sets. In order to find a middle course between runtime performance and image quality the resolution of an (input- and output-) image is chosen to be 300 x 300 pixels where each one is composed of 12 x 12 patches. In case the original image has no square dimensions it is cropped such that the central object of the image is fully preserved. Central image I_C is computed by using the median (expressed by Equation 3.10) in order to ignore outliers. Distance dP is calculated by using the descriptor-method, described in Section 3.2.3. Finally, parameter θ of Equation 3.11 is chosen to be 6 for the probability mapping of APBD and 6 for mapping the probabilities of the patch similarity measurement. Moreover, in Equation 3.11 dP is calculated by choosing a minimum dP of 0 and a maximum dP of 6 800 000. Distance dE is calculated by choosing a minimum dE of 0 and a maximum dE of 750. These values are oriented on the lowest and highest energy values of the distance measurements. A fixed range guarantees a correct ratio among all computed probabilities. For the morphing process the default SIFT flow parameters proposed in [37] are used except γ is set to 0 (as it is done in [72]) such that large vector lengths are not penalized anymore. The parameters used are summarized in Table 4.1.

Figures 4.5, 4.6, 4.8, 4.9, 4.11 and 4.12 present morphed images produced by using 3, 4 and 6 source images where a set of source images originates from one common class. While all source images are grouped at the left hand side, the morphed output is located at the right hand side. The image next to the morphed image has been selected as I_{BestRef} by the algorithm. As can be seen in the given examples, the framework computed examples which are widely aesthetic and coherent. Regions which are considered as outliers (like row 5 / coin image 1 in Figure 4.5 where residuals of an old coin design are visible) are outsourced by the algorithm. In row 1 of Figure 4.12 the glasses and the open mouth of the person is outsourced. Since all images are aligned to I_{BestRef} it can be seen that the morphed image's contours are similar to I_{BestRef}. In contrast, the texture is composed from all source images. Figures 4.7 and 4.10 show arbitrary selected source images meaning that each one originates from a different class. By comparing the morphed images of the different data sets it can be seen that the usage of coin and face images leads to the expected results. Especially the morphed face images represent a neutral expression which forms the best representation of a class. In contrast, morphing images of airplanes does not work as properly as expected.

Image Resolution	I_C	**Distance Metric to I_C**	**θ-Unary**	**θ-Pairwise**	γ
300 x 300 pixels	'median'	descriptor	6	6	0

Table 4.1: Parameters used to obtain T in the qualitative experiments.

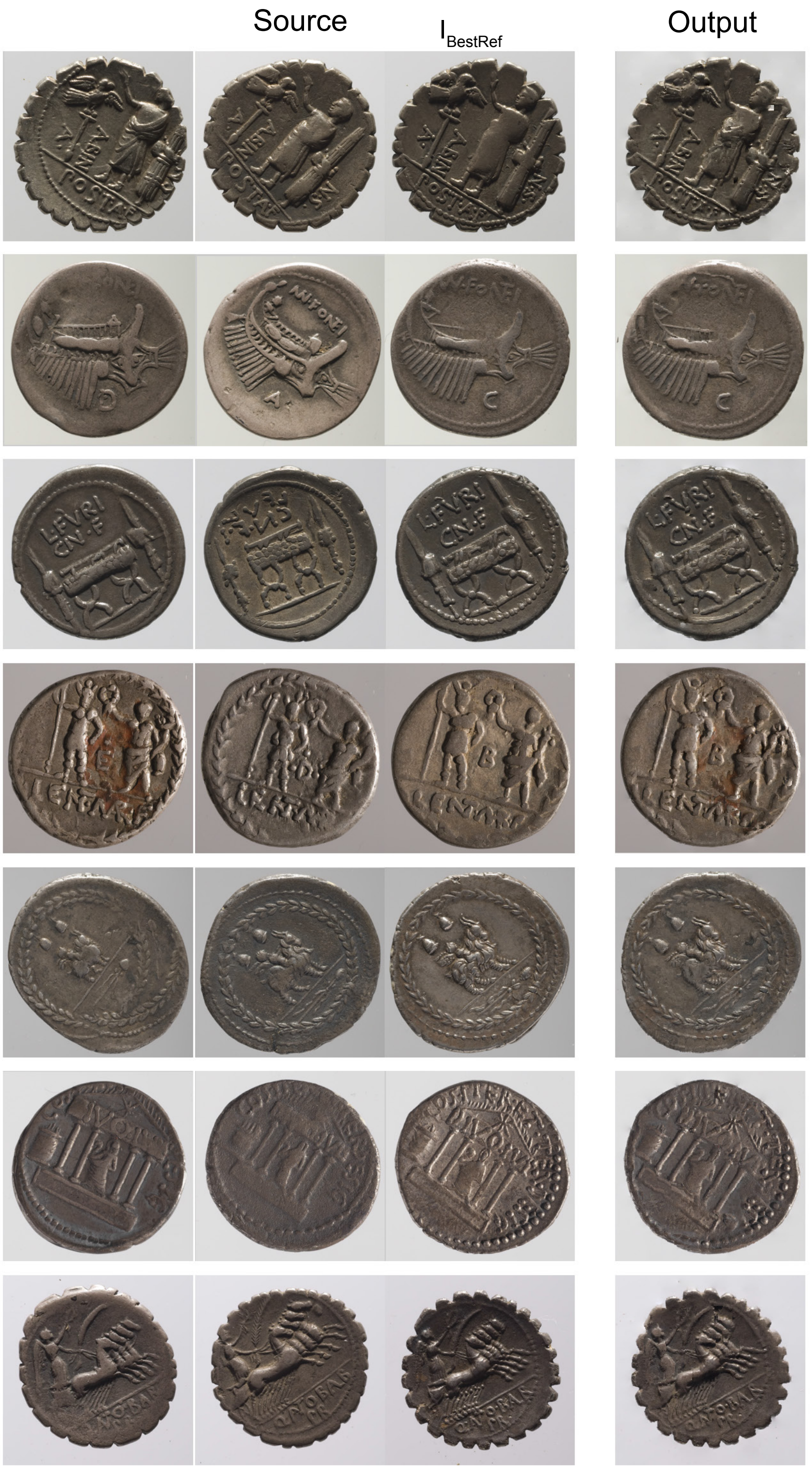


Figure 4.5: A set of morphed coin images using three source images of common classes.

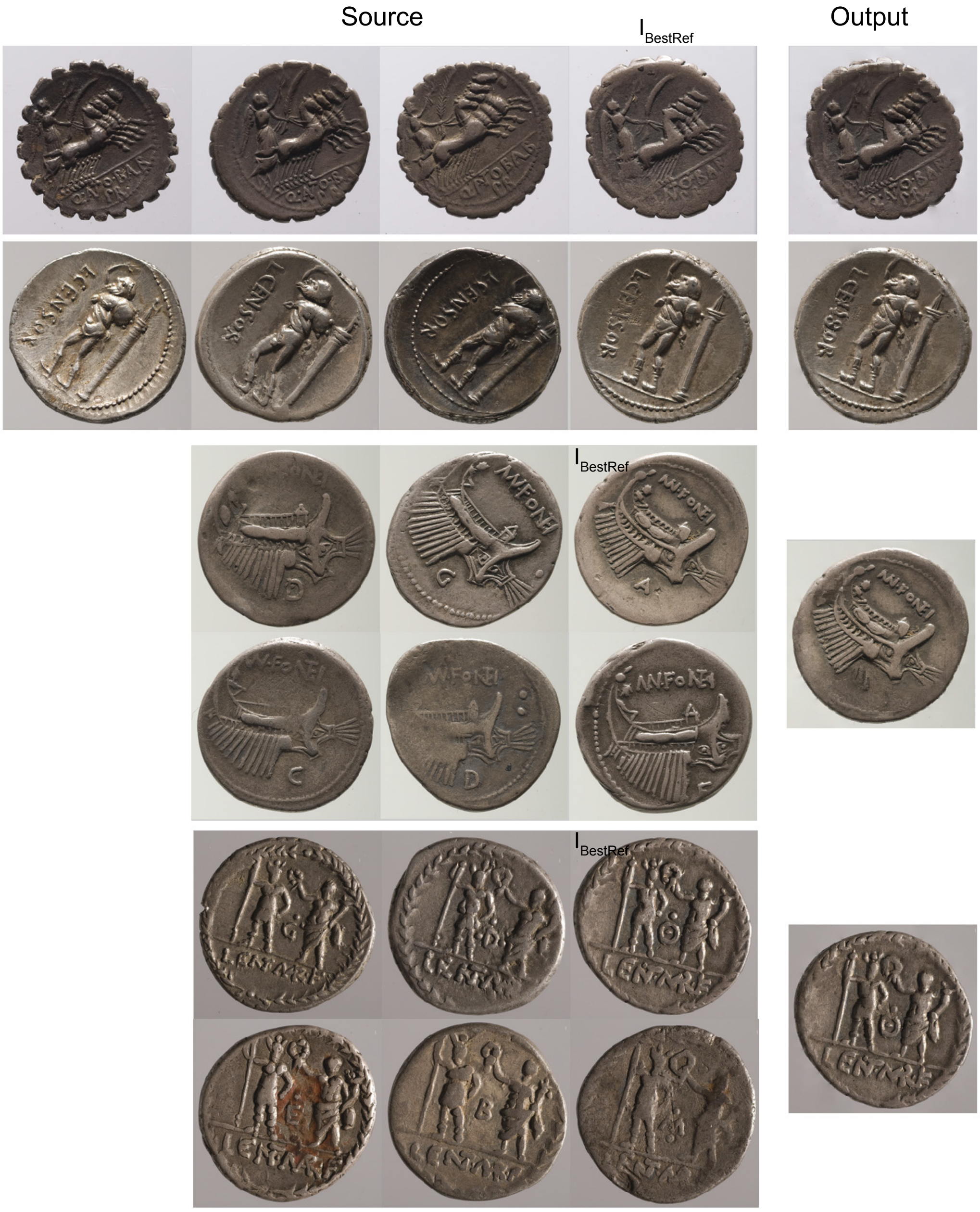


Figure 4.6: A set of morphed coin images using 4 and 6 source images of common classes.

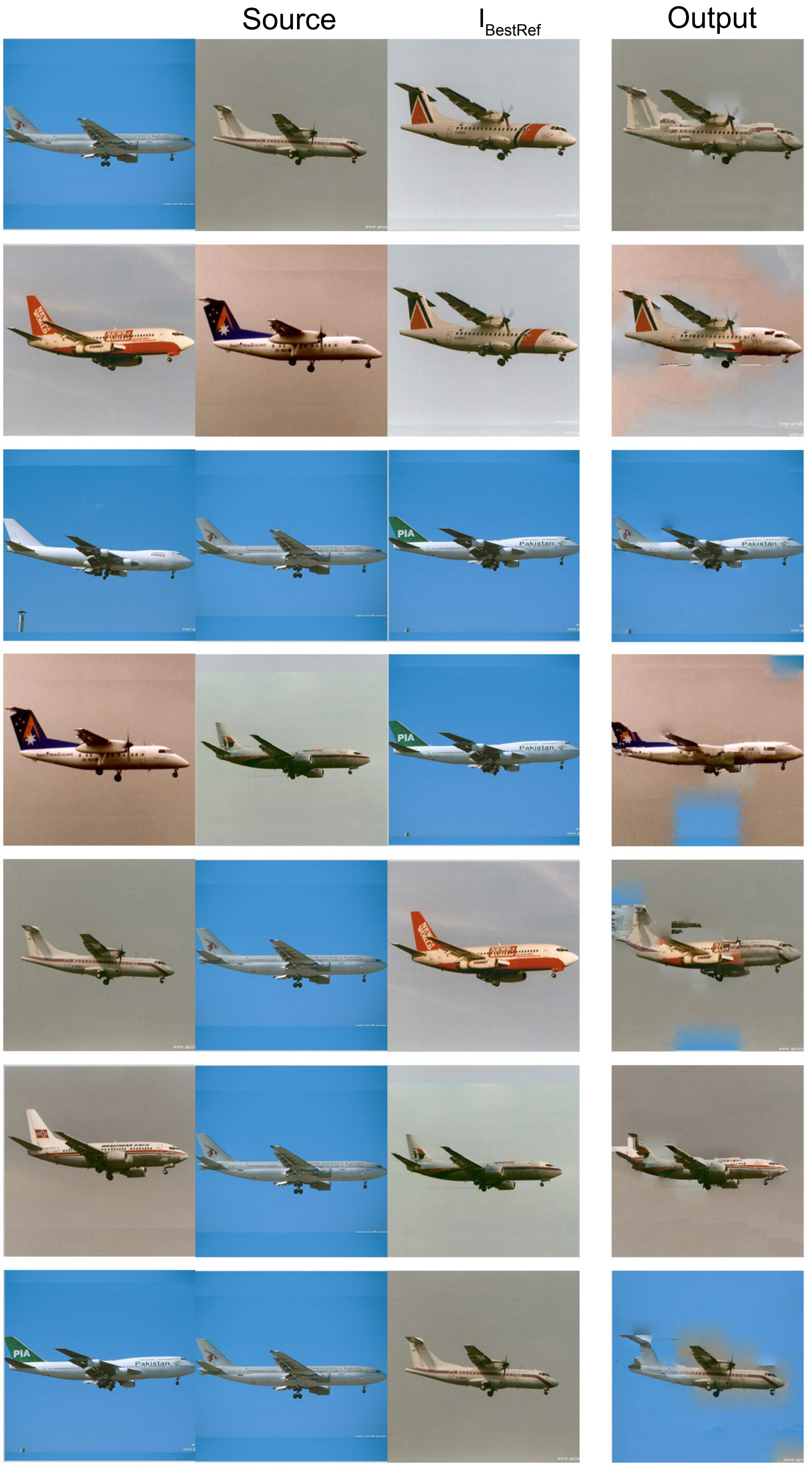


Figure 4.7: A set of morphed airplane images using 3 source images of different classes.

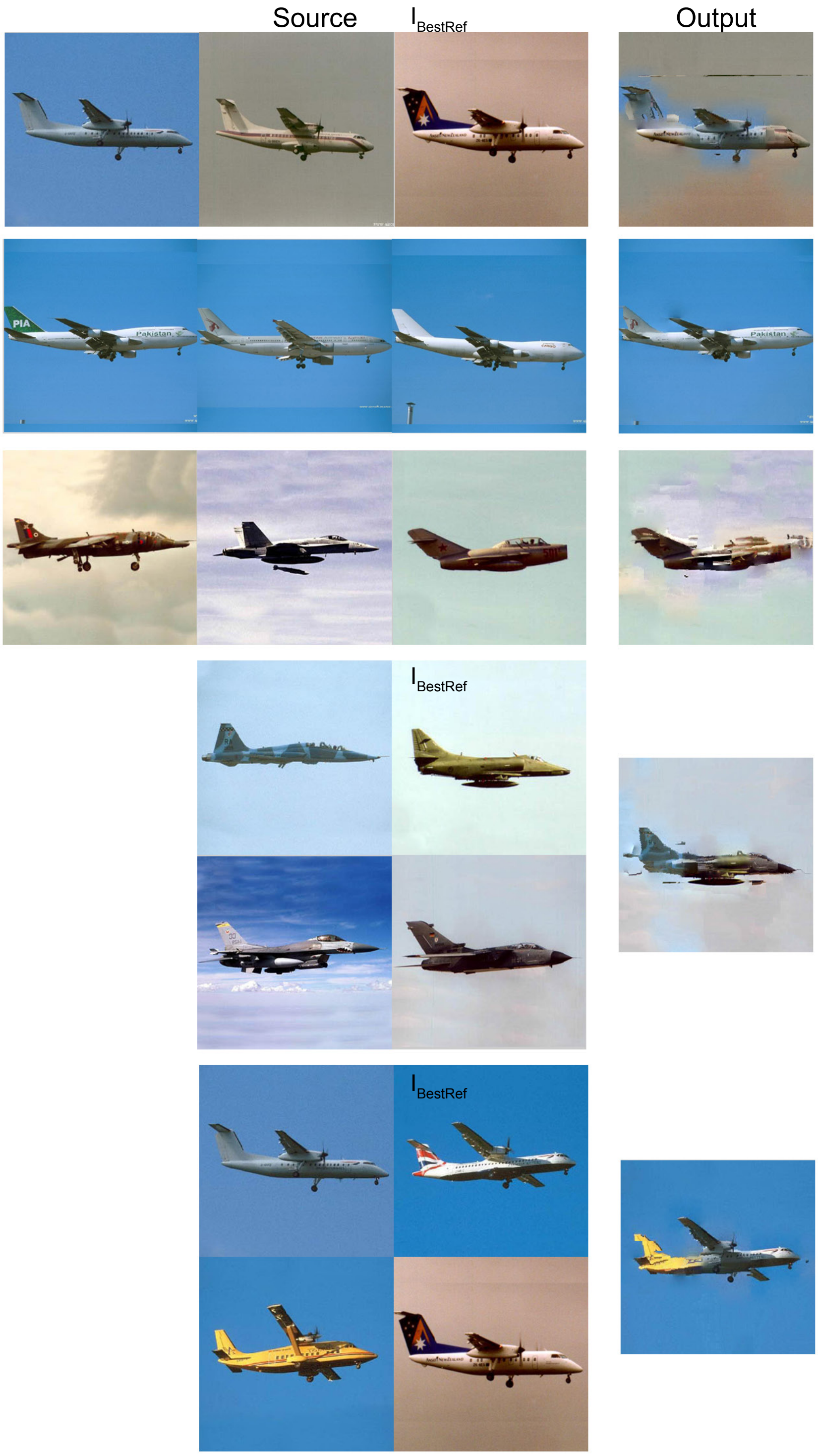


Figure 4.8: A set of morphed airplane images using 3 and 4 source images of common classes.

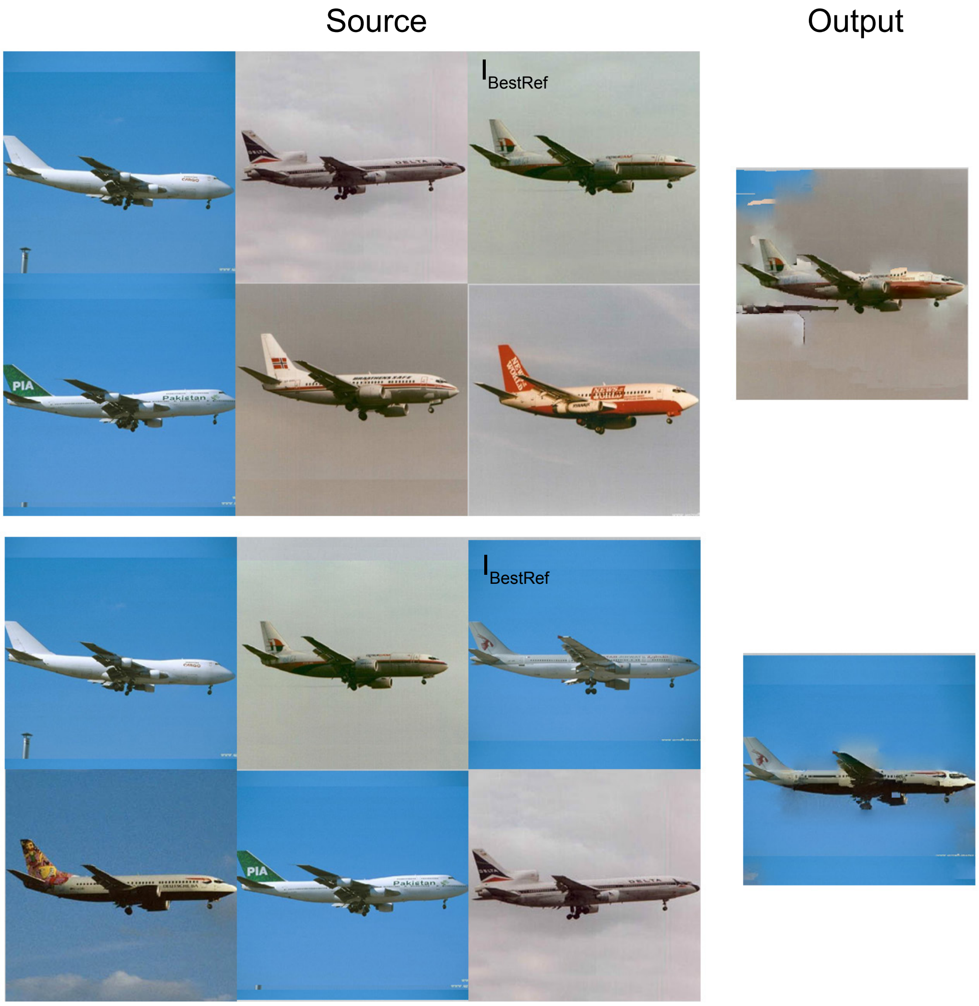


Figure 4.9: A set of morphed airplane images using 6 source images of common classes.

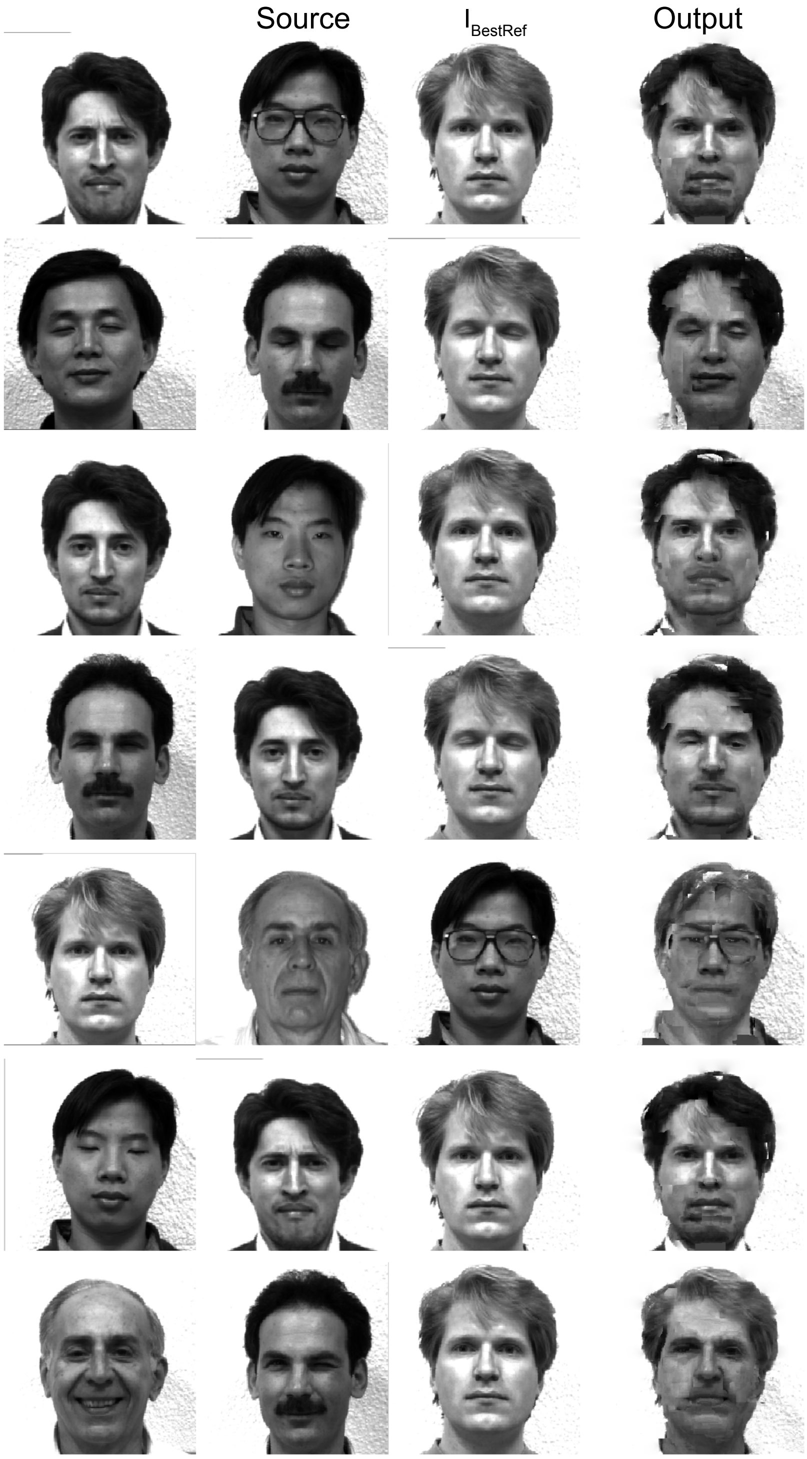


Figure 4.10: A set of morphed face images using 3 source images of different classes.

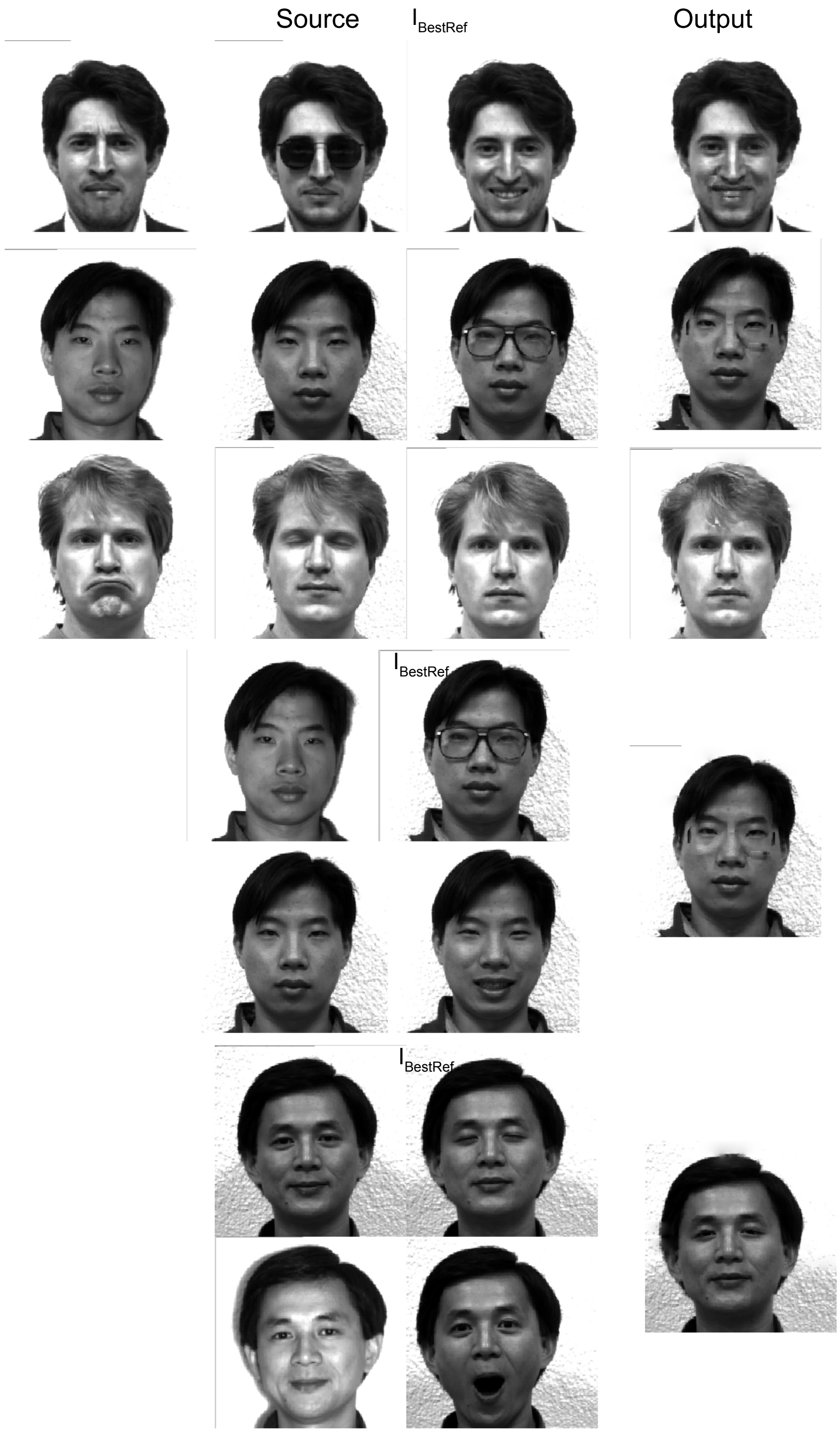


Figure 4.11: A set of morphed face images using 3 and 4 source images of common classes.

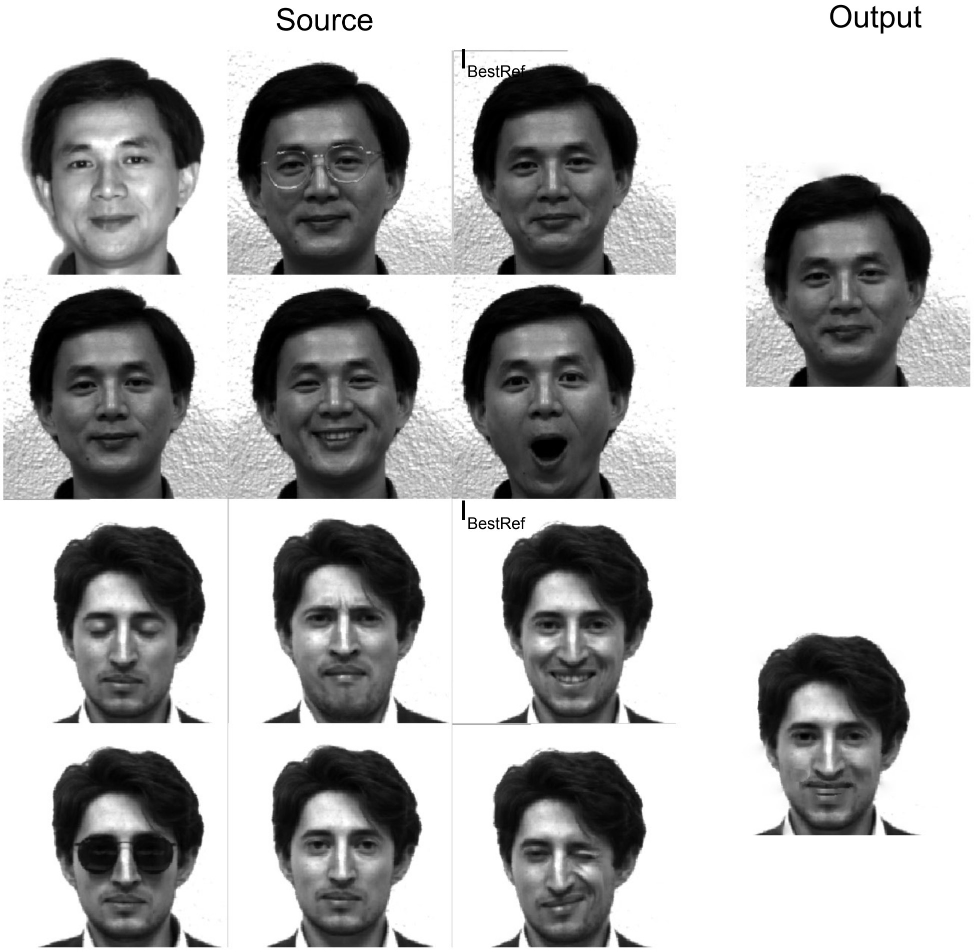


Figure 4.12: A set of morphed face images using 6 source images of common classes.

On the airplane data set artifacts (inappropriated colored regions) are in the background which originate from different background colors of the source images. However, this point of criticism is beyond the scope of this thesis as an uniform background in the input image series is assumed. Figure 4.13 illustrates examples where the algorithm does not work as well as in the previous demonstrations. Artifacts produced by unsatisfactory image registrations fill the background of the morphed images (see rows 3, 4, 5 and 6 in Figure 4.13). In the first row it is shown that the algorithm can not cope with rotated images which is discussed in detail in Section 5. The red squares illustrated in the second row emphasize regions where the image registration was unsatisfactory.

The morphing process is done on a workstation with a quad-core 2.8 GHz Intel Xeon CPU and 8 GB memory. Morphing with 3 source images takes ~7 minutes, 4 source images ~11 minutes and 6 source images ~18 minutes.

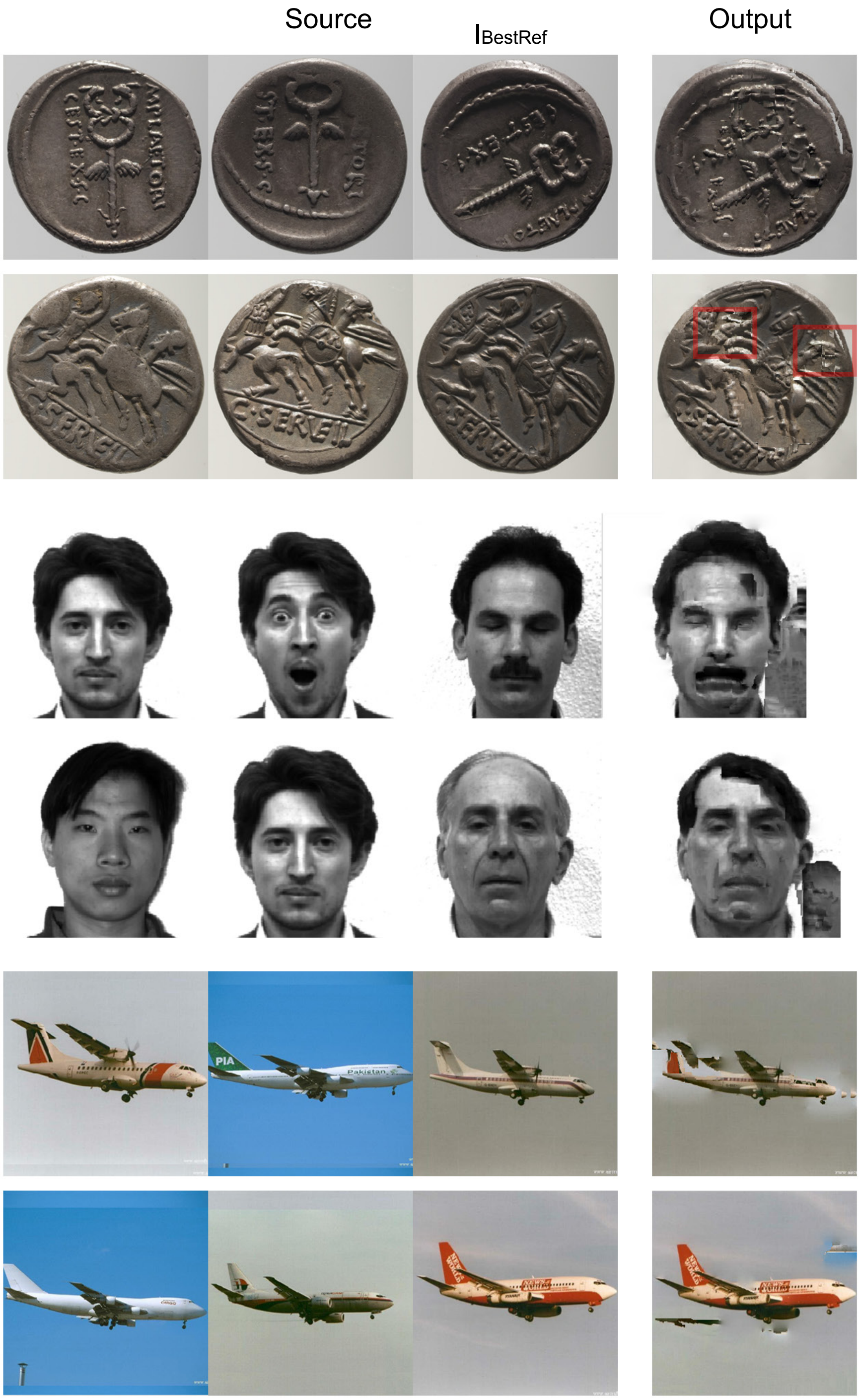


Figure 4.13: A set of morphed coin, face and airplane images where the algorithm does not work properly and artifacts are in the background

4.3 Quantitative Experiments

The quantitative experiments deal with possible applications of the morphed images. They answer the question of how a morphed image can contribute in further computer vision tasks in order to obtain a scientific added value. Since this thesis is realized within the ILAC project[3], the quantitative evaluation is executed by introducing a classification approach of ancient coins which is addressed in several scientific papers such as [2,31,68,69,72]. In detail, this evaluation is an analysis of how well a morphed image represents one common class of given ancient coin images. This method is inspired by [72]. Coin images are obtained from the coin data set described in Section 4.1 which is used as well in a modified form in [72]. All coins depicted in the images are assumed to have similar orientations. In Section 4.3.1 all classification methodologies used in this work are explained. Since the parameters used for the morphing process have to be adjusted in order to deliver the highest classification rate a ROC analysis is done in Section 4.3.2. Finally, the results of the classification task are presented in Section 4.3.3.

4.3.1 Classification Evaluation Schemes

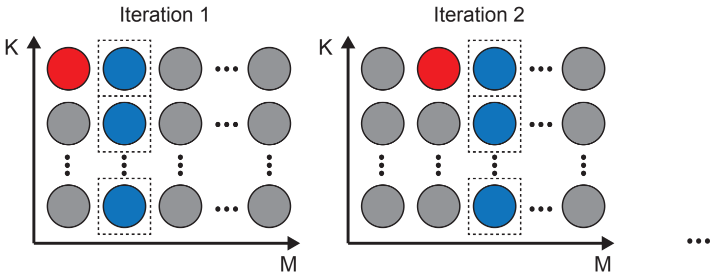
In [72] Zambanini and Kampel exploit the energy term of the SIFT flow algorithm as a similarity metric of images. The energy term is composed of Equations 3.1, 3.2 and 3.3. Once reduced to a minimum, the terms state the visual similarity of two given images. For the coin classification the similarity metric is applied to the coin data set which is divided into different classes. In order to classify one query image among K classes, K training images (one per class) are selected. Subsequently, the energies between the query image and the set of training images are determined in order to assign the query image to the class where the energy is at a minimum. The classification scenario specifies that each image of the data set serves once as query image. This requires the procedure to run through several iterations depending on the number of classes and source images.

The training image is considered as a representation of a class. Depending on typical class features, unified in the training image, the energy term is reduced or increased. This evaluation strategy enables a comparison of several classification tasks where for each task a training image can have different meanings (For a better understanding the different roles of the training image are illustrated in Figure 4.14. All red circles denote the query image and all blue circles the training image(s). The dotted frame indicates the finally selected training image(s)):

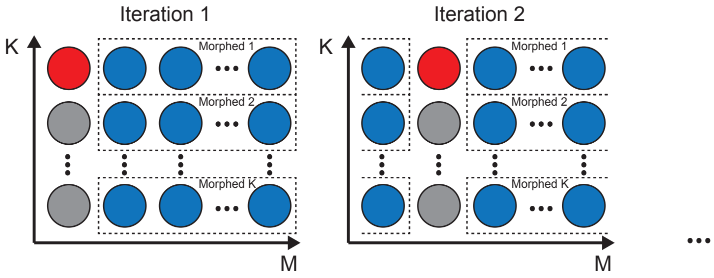
- A single coin image serves as training image.
- The morphed image serves as training image. Denoting M as the number of coin images per class then the morphed image is composed of $M - 1$ images. It has to be ensured that the query image is not used as part of the morphed image.
- Two coin images where the energy is calculated by taking the average of two comparisons per class.
- Two coin images where the lower energy of the comparisons is used.

---

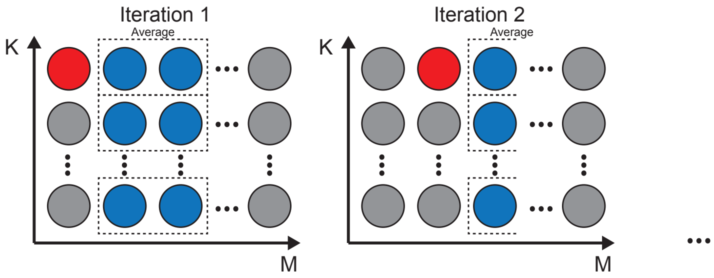
[3] `http://www.caa.tuwien.ac.at/cvl/research/ilac/` (accessed on 03.03.2014)



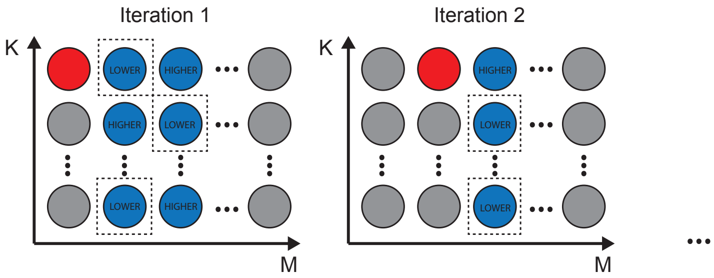
(a) Classification by using a single coin image.



(b) Classification by using a morphed image.



(c) Classification by using the average energy of two coin images.



(d) Classification by using the lower energy of two selected coin images.

Figure 4.14: Illustration of different classification schemes. Red circles denote the query image, blue circles the training image(s). Dotted frames indicates the finally selected training image(s).

The central points of this evaluation are the cases where a single image and a morphed image act as training image. In case of showing a higher classification rate with the latter option the statement can be made that *the representation of a coin class is better by using a morphed image instead of a single image*. In case of using the morphed image as training image, it is necessary to evaluate the best performing parameters on the classification task which help to generate a morphed image.

4.3.2 Parameter Evaluation by Receiver Operating Characteristic Analysis

Since the parameters used for the morphing process have to be adjusted in order to deliver the highest classification rate a ROC curve is calculated [8]. Subsequently, the Area Under Curve (AUC) serves as a performance metric. The greater the AUC the better the performance of the evaluated parameter. In order to prove the independence between the morphing framework and the data set used, for the evaluation of the parameters a different ancient coin data set than for the classification task has to be used. The selection of optimal parameters is based on the assumption of [8] which is also used in [73]:
The better a feature of an image

- the lower the distance between images showing the same scene.
- the higher the distance between images showing different scenes.

Hence, K image pairs showing the same scene, referred to as True Pair (TP) and K image pairs showing different scenes, referred to as False Pair (FP), are compared for a given parameter. An example showing common classes and different classes is given in Figure 4.15. For each query image of a given class obtained from the coin data set one morphed image of the same class and one selected randomly of a different class is used. As similarity metric the energy of SIFT flow is used. The calculated energies are accumulated in a histogram which ranges from the smallest to the highest energy value. The number of bins of the histogram equals K. With the help of the histogram the FP rate and TP rate are confronted to each other by introducing a threshold t

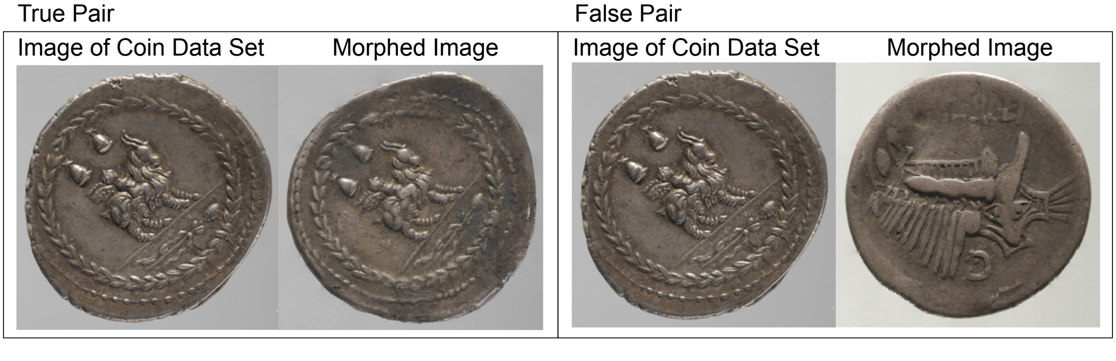


Figure 4.15: Examples of common and different classes.

running from $[0 - K]$. The rates of TP and FP are defined as

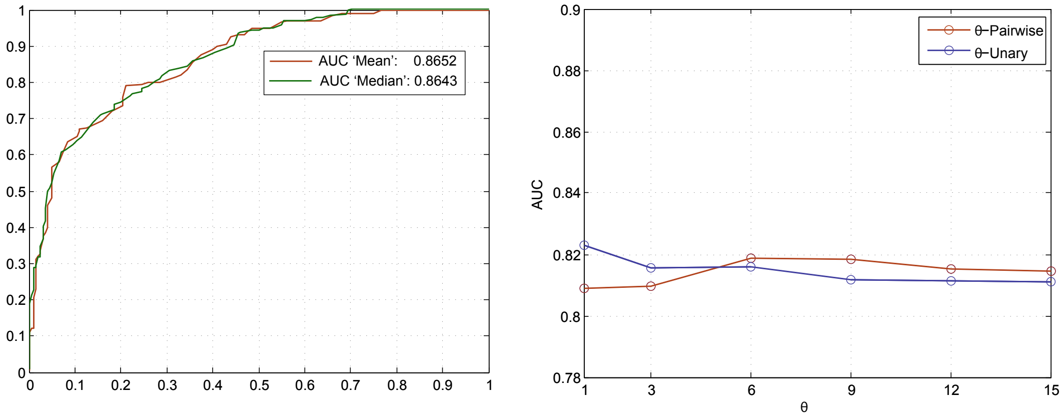
$$\text{TP rate} = \frac{\text{TP}[0, t]}{\text{total TP}} \tag{4.1}$$

$$\text{FP rate} = \frac{\text{FP}[0, t]}{\text{total FP}} \tag{4.2}$$

where TP$[0, t]$ and FP$[0, t]$ denotes the sum of all histogram-values included in the interval $[0, t]$. Due to the assumption of [8] to obtain small distances for TP images and high distances for FP images, for this work a ROC curve is a representation of how well images showing the same scene and images showing different scenes can be classified into these two classes. Consider a ROC curve point $(0, 0)$ represents a TP rate as well as a FP rate of 0. At point $(0, 1)$ the TP rate is 1 and the FP rate is 0 which means in other words that all TP (and thus all FP as well) are classified correctly. Point $(1, 1)$ represents a worst case scenario saying that all images are in the same class. Finally, a greater AUC indicates a clearer separation of two classes [8].
The ROC curve evaluates the following parameters of the morphing algorithm:

- The central image I_C can be generated by computing the 'mean' (see Equation 3.9) or the 'median' of all input images (see Equation 3.10). To obtain a decision which parameter to choose two ROC curves are established and their AUC are compared.
- Parameter *θ-Pairwise* of Equation 3.11 is responsible for the probability mapping of dE (which indicates the distance of a transition between adjacent patches) and *θ-Unary* for the probability mapping of the patch similarity distance dP. In order to obtain an optimized probability mapping, overall 12 ROC curves are established where each parameter takes the values 1, 3, 6, 9, 12 and 15. Moreover, in Equation 3.11 dP is calculated by choosing a minimum dP of 0 and a maximum dP of 6 800 000. Distance dE is calculated by choosing a minimum dE of 0 and a maximum dE of 750. These values are oriented on the lowest and highest energy values of the distance measurements. A fixed range guarantees a correct ratio among all computed probabilities.

The ROC curves for the parameters 'mean' and 'median' are illustrated in Figure 4.16a. The AUC of 0.8652 generated by the 'mean' parameter is slightly larger than 0.8643 generated by the 'median' parameter. The AUC values w.r.t. *θ-Unary* and *θ-Pairwise* are summarized in Figure 4.16b. The generation of e.g. *θ-Unary*'s AUC is done by setting *θ-Pairwise* to a fixed size 6 and vice versa. Thus, *θ-Unary* and *θ-Pairwise* can be selected independently from each other. Considering the progression of both parameters from 1 to 15 it can be noticed that the selection of *θ-Unary* and *θ-Pairwise* does not have a great effect on the classification process. Nevertheless, the configuration with the largest AUC is used for the classification. In this case *θ-Unary* is set to 1 and *θ-Pairwise* takes the value 6. Since the number of classes provided by the data set is limited, values of the parameter evaluation only differ slightly from each other. This means that a higher number of input classes would lead to clearer differentiations. A comparison of the morphed images resulting from quantitative experiments and qualitative experiments show clear differences in their appearances, as can be seen in Figure 4.17. The left column shows resulting images as described in Section 4.2 where the parameter *θ-Unary*



(a) AUC generation dependent on parameters 'mean' and 'median'

(b) The evaluation of θ-*Unary* is done by setting θ-*Pairwise* to the fixed size 6 and vice versa. As can be seen the best performing value is 1 for θ-*Unary* and 6 for θ-*Pairwise*.

Figure 4.16: ROC evaluation of the parameters 'mean', 'med', θ-*Unary* and θ-*Pairwise*.

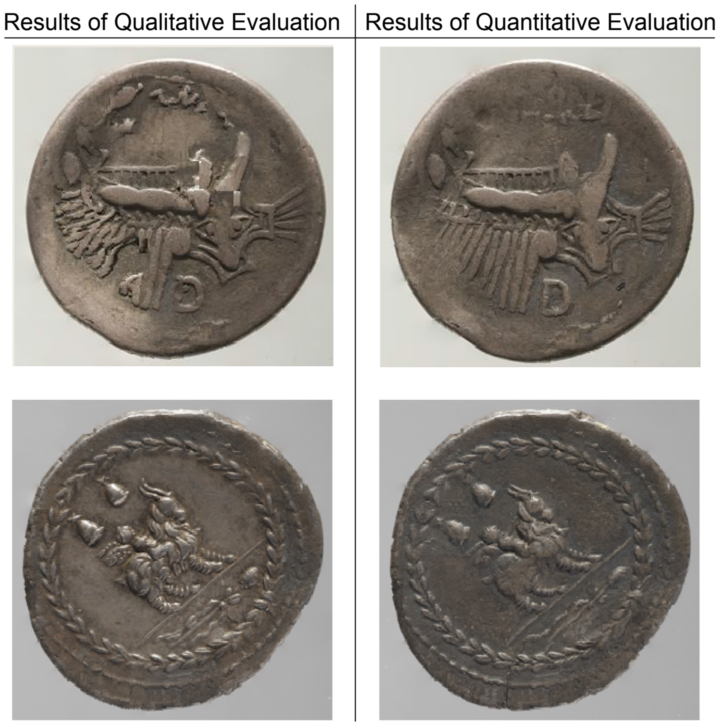

Results of Qualitative Evaluation
Results of Quantitative Evaluation

Figure 4.17: Comparison of visually (obtained from qualitative experiments) and experimentally (obtained from quantitative experiments) best results.

is set to 6 and 'median' is used. The right column shows morphed images with an optimized parameter configuration for the coin classification where θ-*Unary* is set to 1 and the 'mean' parameter is used. Setting θ-*Unary* to 1 implies that only the probability computed for APBD determines if a patch originating from I'_i is added to the finally morphed image or not. Since these probabilities are high within one image the majority of selected patches originate from one image. Parameters used for calculating the SIFT flow as well as calculating distance dP are not included in the parameter evaluation. As proved in [72] the small displacement term is negligible which can be expressed by setting the SIFT flow parameter γ to 0 such that large vector lengths are not penalized anymore [72]. The controlling parameters for the smoothness term are set to $\alpha = 12$ and $\psi = 1200$, the number of pyramid levels to 4 and SIFT features are calculated in a local neighborhood of 12 x 12 pixels [72]. In order to measure dP the descriptor-method is used. A usage of the energy computed by SIFT flow causes a lack of performance. The matching between two 25 x 25 patches takes 27 seconds on a workstation with a quad-core 2.8 GHz Intel Xeon CPU and 8 GB memory.

4.3.3 Classification

For the classification task the coin data set provided consists of 50 classes where each one holds 4 images. In order to select each image once as query image (as the classification scenario specifies), 4 iterations are necessary. Finally, this leads to 200 classified coins. The results of comparing all classification schemes are presented in Table 4.2.

ES	**CC**	**CR**	I_C	**θ-U**	**θ-P**
1: Single image.	179/200	89.5 %	-	-	-
2: Morphed image.	182/200	91.0 %	md	6	1
3: Morphed image.	183/200	91.5 %	mn	6	6
4: Optimized morphed image.	**185/200**	**92.5 %**	**mn**	**6**	**1**
5: Lower energy of two selected images.	192/200	96.0 %	-	-	-
6: Average energy of two images.	195/200	97.5 %	-	-	-

Table 4.2: Classification results. Column description from left to right: Evaluation Scheme, Correct Classified, Classification Rate, Central Image I_C, θ-Unary and θ-Pairwise.

The classification is done on a workstation with a quad-core 2.8 GHz Intel Xeon CPU and 8 GB memory and calculating the energy of one image pair takes ~45 seconds. From overall 200 classified coins the evaluation scheme 1 which uses a single coin image as training image reaches a classification rate of 89.5 %. This rate is clearly outperformed by using morphed images as training image. However, the evaluation scheme using morphed images can be furthermore distinguished depending on the parameter configuration for a morphed image. Evaluation scheme 2 and 3 use a visually aesthetic morphed image as training image and reach a classification rate of 91 % and 91.5 %, respectively. The difference between these visually aesthetic morphed images is that in evaluation scheme 2 θ-Pairwise is set to 1 and 'median' is used and in evaluation scheme 3 θ-Pairwise is set to 6 and 'mean' is used. A morphed image optimized for classification by its parameters (evaluation scheme 3) reaches a classification rate of 92.5 % which means that

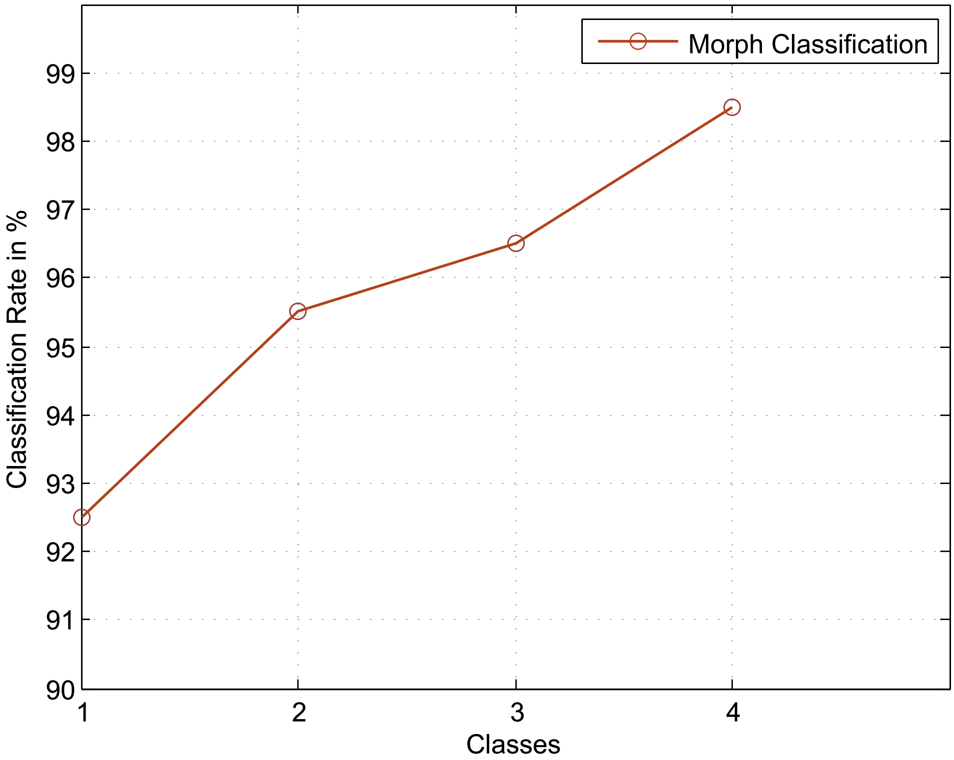


Figure 4.18: Classification rate on y-axis and the class ranking on the x-axis to which a coin image has to be assigned at least in order to be classified correctly.

185 of 200 images are classified correctly. Thus, the hypothesis that *the representation of a coin class is better by using a morphed image instead of a single image* presented in Section 4.3.1 can be regarded as fulfilled. Nevertheless, a classification rate higher than 92.5 % is reached by using two training images where the energy is calculated by either taking the lower (evaluation scheme 5) or the average energy (evaluation scheme 6) of two comparisons per class. However, in contrast to evaluation schemes 2, 3 and 4 (where it is assumed that a morphed image already exists) the runtime is doubled as two comparisons are performed per coin class.

Since Table 4.2 only visualizes the results obtained by using a binary classifier which determines if the coin belongs to a certain class or not in the following a detailed examination of evaluation scheme 4 is presented. In Figure 4.18 not only the first class to which a coin is assigned is considered but the first four classes. The better the ranking of a class the higher the probability that a coin image is assigned to this class. Considering a coin image as correctly classified if the query image belongs either to the first or to the second class, then the classification rate is 96 %. Furthermore, it can be noticed that for evaluation scheme 4 198 classified coin images are ranked within the first four classes.

4.4 Discussion

The results of this work show that it is possible to summarize visual information from multiple ancient coin images. However, it has to be distinguished if a coin is morphed in order to obtain an aesthetic image or to achieve an optimized classification. As shown in Section 4.2, the proposed algorithm generates morphed images in a complete and coherent way resulting in an aesthetic coin image. Using the morphed coin image as a training image, in the morphing process the probability if a patch is added or not remains the same for all patches since **θ-Unary** is set to 1. Therefore, it is not necessary anymore to compute the joint probability of the patch similarity measurement and the APBD. The only remaining probability computed for APBD determines if a patch originating from I_i' is added to the finally morphed image or not. Since these probabilities are high within one image the majority of selected patches originate from one image. In summary, in order to improve a classification of ancient coins it is not necessary to summarize visual information in form of patches from multiple ancient coins, but rather to find an existing coin image which exhibits the common contours of a class (represented as I_{BestRef}).

4.5 Summary

In this chapter the evaluation of the system is presented which is divided into qualitative and quantitative experiments. Qualitative experiments are focusing on the visual appearance of the morphed image. In order to demonstrate the flexibility of the morphing framework it is applied to three different data sets: coin images of the Roman Republican coinage, airplane images which are all photographed sideways and a gray scale image data set illustrating different appearances of human faces. In order to show applications using the morphing framework, quantitative experiments are carried out. For this experiments the coin data set is used. Furthermore, a classification task is described where a morphed image serves as training image in order to decide to which class a coin image has to be assigned. The energy term of the SIFT flow algorithm is employed as similarity metric of images. In order to obtain an optimized training image the best parameter configuration of the morphing framework is found by generating ROC curves for a given parameter. The data set comprises 50 classes, each one holding 4 images. If each image of the data set is classified once, 200 class assignments have to be done. This procedure answers the question of how well a morphed image is able to represent an image class. With a classification rate of 92.5 % it can be said that using a morphed image clearly outperforms the case of a single training image. The rate is increased by 3 %. However, the best classification rate of 97.5 % is obtained by taking the average energy of two training images but this method implies a higher runtime.

CHAPTER 5

Conclusion

The thesis is concerned with the summarization of visual content from multiple source images. Visual regions which match to several other images are considered as valuable and have to be preserved while non-matchable regions have to be eliminated. As output the algorithm computes a single morphed image where the content is composed in a coherent and complete way. This means that as much as possible common regions of all source images are summarized in one image and at the same time the generated image appears aesthetic without exhibiting any disturbing artifacts.

In this work this procedure is called multi-image morphing. Initially, image registration methods are investigated, since this step forms the basis of the morphing system. Registration provides a possibility to bring given images into a relationship to each other. Image registration methods are used for several computer vision tasks such as view morphing, image fusion, image stitching or image completion. For this work SIFT flow is used in order to register images to each other. SIFT flow is based on the SIFT algorithm and allows a dense approximate correspondence search. The result is a displacement field holding concrete displacements for each pixel. Subsequently, a so called best reference image is chosen by finding the minimum of pixel displacements over all corresponding images. A central image is considered as a container holding visually valuable regions. This is realized by calculating the mean or the median of all corresponding pixel values. According to the central image, the morphing process reproduces a visually similar, sharp output by comparing image patches to the central image. An optimization is done by firstly exploiting a MRF whose objective is to find the best combination of similar patches compared to the central image and the slightest visual difference between patches of the output image. Secondly, the solution of a Poisson equation smooths the morphed image such that any boundaries or disturbing seams become invisible.

The evaluation of the algorithm is divided into qualitative and quantitative experiments: firstly, the qualitative experiments show examinations of the morphing results according to their visual appearance. The objective is to select a parameter configuration in order to obtain visually aesthetic images. All images used are obtained from three different data sets: coin images of the Roman Republican coinage, airplanes which are all photographed sideways and gray scale images showing different appearances of human faces. It is shown that the proposed framework is not only applicable to ancient coin images but also on the two further mentioned data sets. Images are morphed from each data set where 3, 4 and 6 source images are used, originating from both common and different classes. Since the term „aesthetic“ is a subjective quality measurement, secondly quantitative experiments are introduced. These experiments are dealing with a possible application of the morphed images. They answer the question of how a morphed image can contribute in further computer vision tasks. For this purpose, a morphed image is used as training image in order to decide to which class a coin image has to be assigned. The classification scheme uses the energy term of the SIFT flow algorithm as similarity metric of images. Once reduced to a minimum the terms state the visual similarity of two given images. The similarity metric is conducted on the coin data set which is divided into 50 different classes where each one holds 4 images. In order to classify one query image among those classes, 50 training images (one per class) are selected. Subsequently the energies between the query image and the set of training images are determined in order to assign the query image to the class where the energy is at a minimum. In this context the term 'training image' can take different meanings: a training image represents either a single coin image, a morphed image, two coin images where the energy is calculated by taking the average of two comparisons per class, or two coin images where the lower energy of the comparisons is used. In case of using the morphed image as training image in the classification process, an optimized parameter configuration has to be found. This is realized by considering ROC curves established by confronting TP (images of the same class) and FP (images of different classes). With an optimized parameter configuration it is shown that the hypothesis: *the representation of a coin class is better by using a morphed image instead of a single image*, can be confirmed. The classification rate is increased by 3 % compared to the usage of a single image as training image. However, using a morphed image as training image is outperformed by taking either the lower energy or the average energy as classification scheme.

The results of this work show that it is possible to summarize visual information from multiple ancient coins in one image but it has to be distinguished if a coin is morphed in order to obtain an aesthetic coin image or to generate an optimized classifier. The parameter configuration of the morphing framework changes according to its purposes. In case of using it as classifier the probability of all patches (which are candidates to get added to the morphed image) has to be set to 1. Thus, joint probabilities do not have to be computed anymore by a MRF. Only the probabilities of the APBD determines if a patch is selected or not. From this follows that the majority of patches originating from a common source image are selected. In order to improve a classification of ancient coins it is not necessary to summarize visual information in form of patches from multiple ancient coins, but rather to find an existing coin image which exhibits common contours of a class.

This work shows a completely new approach of morphing multiple images. Strengths of this system are that no restrictions regarding the number of source images exist which is a difference compared to related approaches. Furthermore, arbitrary types of objects illustrated in the images can be morphed by the system. The more source images used for the morphing task the better the representation of a class. However, with an increasing number of source images the runtime is increased exponentially.

Limitations

A drawback of the system is that it can not cope with different orientations of source images. This case is not considered in the registration process of the morphing framework and implies that all source images are assumed to be oriented in a similar direction. Furthermore, artifacts can be produced by an unsatisfactory image registration of the source images. They can appear for instance as unsolicited regions located at the background of a morphed image. The focus of this system is on morphing coin images which exhibit homogeneous background colors and structures. A possible reason for artifacts is a inhomogeneous background (e.g. clouds behind airplanes or shadows behind faces). Moreover, the patches used are rectangular with straight boundaries and they are arranged in a static way which might clip important image content in course of the patch selection.

Future Work

In order to extend and improve the system three concrete suggestions could be made. Firstly, once the images are aligned to each other rectangular image patches could be substituted by super-pixels. In this scope super-pixels are image regions defined by irregular borders. As proposed in [28] an optimal seam between these borders could be calculated in order to improve the transitions between neighboring regions. In contrast to the current system, super-pixels' boundaries can be adapted more precisely to the images' content which leads to smoother transitions. Since the focus of the system is on morphing coin images where a homogeneous background is assumed a segmentation in foreground and background can be realized in order to achieve an improvement in image registration. Moreover, the SIFT flow algorithm allows to use rotational invariant features. By including that kind of SIFT features into the morphing framework correct pixel correspondences could be found even though the images' content show different orientations. Finally, the evaluation of the parameter configuration would be improved by including more than 200 coin classes in order to compute a ROC curve. The increase of input classes would deliver better distinguishable AUC values.

The proposed system can be used for any computer vision task where a summarized representation of one class is necessary. This might for instance be the case when one image is listed instead of a whole image set due to cost or space limitations. Furthermore, this might be the case when the runtime of a classification task should stagnate but in the same time the quality of the class representation is increased in the training image. The morphing framework can be used for computer vision tasks where only one image is unstable and the summarization of multiple data leads to a more stable representation. This might be used in digital satellite images comprising images where clouds occlude land areas. By morphing multiple images containing common scenes of the land area disturbing clouds can be outsourced automatically.

Bibliography

[1] T.S. Anand, Kumaravelu Narasimhan, and P. Saravanan. Performance evaluation of image fusion using the multi-wavelet and curvelet transforms. In *Proc. of International Conference on Advances In Engineering, Science and Management (ICAESM)*, pages 121–129, 2012.

[2] Hafeez Anwar, Sebastian Zambanini, and Martin Kampel. A bag of visual words approach for symbols-based coarse-grained ancient coin classification. In *Annual Workshop of the Austrian Association for Pattern Recognition (AAPR)*, http://arxiv.org/abs/1304.6192, 2013.

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3), 2009.

[4] Connelly Barnes, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 29–43, 2010.

[5] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In *Proc. of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 35–42, 1992.

[6] Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):711–720, 1997.

[7] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(4):509–522, 2002.

[8] Matthew Brown, Gang Hua, and Simon A. J. Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(1):43–57, 2011.

[9] Matthew Brown and David G. Lowe. Invariant features from interest point groups. In *Proc. of British Machine Vision Conference (BMVC)*, pages 1–10, 2002.

[10] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision (IJCV)*, 61(3):211–231, 2005.

[11] Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[12] Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. In *Proc. of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 569–577, 2003.

[13] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proc. of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 279–288, 1993.

[14] Michael H. Crawford. *Roman Republican Coinage*, volume 2. Cambridge University Press, 1974.

[15] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image Melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics*, 31(4):82:1–82:10, 2012.

[16] Warren Esty. The theory of linkage. *Numismatic Chronicle*, 150:205–222, 1990.

[17] Shiaofen Fang, Rajagopalan Srinivasan, Raghu Raghavan, and Joan T. Richtsmeier. Volume morphing and rendering - an integrated approach. *Computer Aided Geometric Design (CAGD)*, 17(1):59–81, 2000.

[18] Allan Daniel Finistauri and Fengfeng (Jeff) Xi. Type synthesis and kinematics of a modular variable geometry truss mechanism for aircraft wing morphing. In *Proc. of International Conference on Reconfigurable Mechanisms and Robots (ReMAR)*, pages 478–485, 2009.

[19] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*, volume 3. Spink and Son, 2009.

[20] Mark Goodman. *Numismatic Photography*. Zyrus Press, 2009.

[21] Philip Grierson. *Numismatics*. Oxford University Press, 1975.

[22] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[23] Reinhold Huber, Herbert Ramoser, Konrad J. Mayer, Harald Penz, and Michael Rubik. Classification of coins using an eigenspace approach. *Pattern Recognition Letters (PRL)*, 26(1):61–75, 2005.

[24] Reinhold Huber-Mörk, Maia Zaharieva, and H. Czedik-Eysenberg. Numismatic object identification using fusion of shape and local descriptors. In *Proc. of International Symposium on Visual Computing (ISVC)*, pages 368–379, 2008.

[25] Reinhold Huber-Mörk, Sebastian Zambanini, Maia Zaharieva, and Martin Kampel. Identification of ancient coins based on fusion of shape and local features. *Machine Vision and Applications (MVA)*, 22(6):983–994, 2011.

[26] Jiaya Jia and Chi-Keung Tang. Eliminating structure and intensity misalignment in image stitching. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 1651–1658, 2005.

[27] Jiaya Jia and Chi-Keung Tang. Tensor voting for image correction by global and local intensity alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(1):36–50, 2005.

[28] Jiaya Jia and Chi-Keung Tang. Image stitching using structure deformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(4):617–631, 2008.

[29] Michael J. Jones and Tomaso Poggio. Multidimensional morphable models. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 683–688, 1998.

[30] Martin Kampel, Reinhold Huber-Mörk, and Maia Zaharieva. Image-based retrieval and identification of ancient coins. *IEEE Intelligent Systems*, 24(2):26–34, 2009.

[31] Martin Kampel and Maia Zaharieva. Recognizing ancient coins based on local features. In *Proc. of International Symposium on Visual Computing (ISVC)*, pages 11–22, 2008.

[32] Johannes Kopf, Wolf Kienzle, Steven M. Drucker, and Sing Bing Kang. Quality prediction for image completion. *ACM Transactions on Graphics*, 31(6):131:1–131:8, 2012.

[33] Aaron W. F. Lee, David P. Dobkin, Wim Sweldens, and Peter Schröder. Multiresolution mesh morphing. In *Proc. of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 343–350, 1999.

[34] Seungyong Lee, Kyung-Yong Chwa, and Sung Yong Shin. Image metamorphosis using snakes and free-form deformations. In *Proc. of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 439–448, 1995.

[35] Seungyong Lee, George Wolberg, Kyung-Yong Chwa, and Sung Yong Shin. Image metamorphosis with scattered feature constraints. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2(4):337–354, 1996.

[36] Seungyong Lee, George Wolberg, and Sung Yong Shin. Polymorph: Morphing among multiple images. *IEEE Computer Graphics and Applications*, 18(1):58–71, 1998.

[37] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5):978–994, 2011.

[38] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman. Sift flow: Dense correspondence across different scenes. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 28–42, 2008.

[39] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[40] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[41] Ren C. Luo, Ching-Chung Kao, and Yen-Chang Wu. Hybrid discriminative visual object tracking with confidence fusion for robotics applications. In *Proc. of International Conference on Intelligent Robots and Systems (IROS)*, pages 2965–2970, 2011.

[42] Russell A. Manning and Charles R. Dyer. Interpolating view and scene motion by dynamic view morphing. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1388–1394, 1999.

[43] Simon Masnou and Jean-Michel Morel. Level lines based disocclusion. In *Proc. of IEEE International Conference on Image Processing (ICIP)*, pages 259–263, 1998.

[44] Mark Mudge, Jean-Pierre Voutaz, Carla Schroer, and Marlin Lum. Reflection transformation imaging and virtual representations of coins from the hospice of the grand st. bernard. In *Proc. of International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)*, pages 29–39, 2005.

[45] Michael Nölle, Harald Penz, Michael Rubik, Konrad J. Mayer, Igor Holländer, and Reinhard Granec. Dagobert - a new coin recognition and sorting system. In *Proc. of International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 329–338, 2003.

[46] Sebastian Nowozin and Christoph H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4):185–365, 2011.

[47] Jindrich Parus, Ivana Kolingerová, and Martina Málková. Multimorphing: A tool for shape synthesis and analysis. *Advances in Engineering Software*, 40(5):323–333, 2009.

[48] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, 2003.

[49] Matthew R. Peters, Leonard Mcmillan, and R. Peters. Multidimensional image morphs: Construction and user interface. Master's thesis, Massachusetts Institute of Technology, 2003.

[50] Md. Iqbal Quraishi, Goutam Das, Krishna Gopal Dhal, and Pratiti Das. Classification of ancient coin using artificial neural network. *International Journal of Computer Applications (IJCA)*, 62(18):6–9, 2013.

[51] Seon-Min Rhee, Jongmoo Choi, and Ulrich Neumann. Stereoscopic view synthesis by view morphing. In *Proc. of International Symposium on Visual Computing (ISVC)*, pages 924–933, 2008.

[52] Mohammad Amin Sadeghi, Seyyed Mohammad Mohsen Hejrati, and Niloofar Gheissari. Poisson local color correction for image stitching. In *Proc. of International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 275–282, 2008.

[53] Steven M. Seitz and Charles R. Dyer. View morphing. In *Proc. of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 21–30, 1996.

[54] Eli Shechtman, Alex Rav-Acha, Michal Irani, and Steven M. Seitz. Regenerative morphing. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 615–622, 2010.

[55] Alla Sheffer and Vladislav Kraevoy. Pyramid coordinates for morphing and deformation. In *Proc. of International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 68–75, 2004.

[56] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[57] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proc. of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 251–258, 1997.

[58] Laurens J. P. van der Maaten and Paul J. Boon. A fast system for reliable coin classification. In *MUSCLE CIS Coin Competition Workshop*, pages 7–17, 2006.

[59] Laurens J. P. van der Maaten and Eric O. Postma. Towards automatic coin classification. In *Digital Cultural Heritage - Essential for Tourism*, volume 211, pages 19–26, 2006.

[60] Thomas Vetter, Michael J. Jones, and Tomaso Poggio. A bootstrapping algorithm for learning linear models of object classes. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 40–46, 1997.

[61] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(3):463–476, 2007.

[62] George Wolberg. *Digital image warping*. IEEE, 1990.

[63] George Wolberg. Image morphing: a survey. *The visual computer*, 14(8-9):360–372, 1998.

[64] Jiangjian Xiao and Mubarak Shah. From images to video: View morphing of three images. In *Proc. of Conference on Vision, Modeling, and Visualization (VMV)*, pages 495–502, 2003.

[65] Jiangjian Xiao and Mubarak Shah. Tri-view morphing. *Computer Vision and Image Understanding (CVIU)*, 96(3):345–366, 2004.

[66] Dong Xu, Hongxin Zhang, Qing Wang, and Hujun Bao. Poisson shape interpolation. *Graphical Models*, 68(3):268–281, 2006.

[67] Yong Yang. Multimodal medical image fusion through a new dwt based technique. In *Proc. of International Conference on Bioinformatics and Biomedical Engineering (ICBBE)*, pages 1–4, 2010.

[68] Maia Zaharieva, Reinhold Huber-Mörk, Michael Nölle, and Martin Kampel. On ancient coin classification. In *Proc. of International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)*, pages 55–62, 2007.

[69] Maia Zaharieva, Martin Kampel, and Sebastian Zambanini. Image based recognition of ancient coins. In *Proc. of International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 547–554, 2007.

[70] Sebastian Zambanini and Martin Kampel. Robust automatic segmentation of ancient coins. In *Proc. of International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 273–276, 2009.

[71] Sebastian Zambanini and Martin Kampel. Automatic coin classification by image matching. In *Proc. of International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)*, pages 65–72, 2011.

[72] Sebastian Zambanini and Martin Kampel. Coarse-to-fine correspondence search for classifying ancient coins. In *ACCV Workshop on e-Heritage*, pages 25–36, 2012.

[73] Sebastian Zambanini and Martin Kampel. Evaluation of low-level image representations for illumination-insensitive recognition of textureless objects. In *International Workshop on Multimedia for Cultural Heritage (ICIAP)*, pages 71–80, 2013.

[74] Sebastian Zambanini, Martin Kampel, and Mario Schlapke. On the use of computer vision for numismatic research. In *Proc. of International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)*, pages 17–24, 2008.

[75] Sebastian Zambanini, Albert Kavelar, and Martin Kampel. Improving ancient roman coin classification by fusing exemplar-based classification and legend recognition. In *International Workshop on Multimedia for Cultural Heritage (ICIAP)*, pages 149–158, 2013.

[76] Sebastian Zambanini, Mario Schlapke, Martin Kampel, and Andreas Müller. Historical coins in 3d: Acquisition and numismatic applications. In *Proc. of International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)*, pages 49–52, 2009.

[77] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.

Nomenclature

1D	1 Dimensional.
3D	3 Dimensional.
APBD	Adjacent Patch-Boundary Disparity.
AUC	Area Under Curve.
BDS	Bidirectional Similarity.
BoW	Bag Of Visual Words.
CT	Computer Tomography.
DCSM	Deviation from circular shape matching.
DOG	Difference-of-Gaussian.
DOP	Double Optimal Partitions.
FFD	Free Form Deformation.
FP	False Pair.
FWF	Austrian Science Fund.
ILAC	Image-based Classification of Ancient Coins.
MAP	Maximum a-posteriori.
MR	Magnetic Resonance.

MRF	Markov Random Field.
NNF	Nearest-Neighbor Field.
PCA	Principal Component Analysis.
PTM	Polynomial Texture Mapping.
ROC	Receiver Operating Characteristic.
SAD	Sum of Absolute Difference.
SIFT	Scale Invariant Feature Transform.
SOP	Single Optimal Partitions.
SSD	Sum of Squared Distances.
TP	True Pair.
UGM	Undirected Graphical Models.